Application Operations Management

Service Overview

Issue	01
Date	2025-01-07

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 What Is AOM?

Application Operations Management (AOM) is a one-stop, multi-dimensional O&M management platform for cloud applications. It integrates observable data sources, such as Cloud Eye, Log Tank Service (LTS), Application Performance Management (APM), real user experience, and backend link data. It also provides one-stop observability analysis solutions. With AOM, you can detect faults in a timely manner, monitor applications, resources, and services in real time, and improve automated O&M capability and efficiency.

- Hosting & Running

  AOM seamlessly interconnects with multiple upper-layer O&M services. It can quickly collect metric data from services such as ServiceStage, FunctionGraph, and Cloud Service Engine (CSE), and display them in real time.

- Observability Analysis

  Provides observable analysis capabilities such as exception detection, historical data analysis, performance analysis, correlation analysis, and scenario-based analysis through container/Prometheus monitoring based on multi-scenario, -layer, and -dimensional metric data.

- Collection Management

  Manages plug-ins centrally and issue instructions for operation such as script delivery and execution.

- Openness

  Supports reporting of native Prometheus Query Language (PromQL) data, data reporting through APIs, and data viewing through Grafana.

2 Advantages

- **Compatibility and openness**

  AOM supports various open-source protocols, opens O&M data query APIs and collection standards, and provides fully hosted, O&M-free, and cost-efficient cloud native monitoring capabilities.

- **Ready-to-use**

  You can connect applications to AOM without changing code. Data can be collected in a non-intrusive way.

- **Full-stack integrated monitoring**

  AOM monitors data of clients, servers, and cloud products. It supports data discovery and display, and reports alarms when there are exceptions. It implements integrated monitoring from top to bottom and from the frontend to the backend.

- **Precise alarm reporting**

  AOM has a unified alarm system, covering metric, log, and event alarms. It provides alarm noise reduction policies, such as grouping, suppression, and silence. It also supports alarm notification and subscription, so that you can easily cope with alarm storms and detect and clear alarms.

- **Unified visualization**

  Multiple data sources can be monitored and analyzed in the same dashboard. They are displayed in various graphs (such as line and digit graphs), helping you better monitor resources, learn about trends, and make decisions.

3 Application Scenarios

Improving User Experience

Pain Points

Optimal user experience has become the core competitiveness of Internet enterprises. They strive to monitor real user experience, reduce churn rate, and improve user conversion rate.

Solutions

AOM analyzes the complete process (user request > server > database > server > user request) of transactions in real time, enabling enterprises to better monitor user experience. For transactions with poor experience, AOM locates problems through topology and tracing.

- Real user monitoring (RUM) monitors page performance, JS error requests, API requests, and service operations metrics (such as PV and UV) in real time.
- User session tracing locates slow requests, loading, and interactions that affect user experience and monitors user usage in real time.
- Page loading performance analysis provides metrics (such as the time to first frame, white screen time, and interaction time), helping you restore user experience and locate the causes of slow access.

Maintaining Containers

Pain Points

Prometheus is ideal for monitoring containers. Since self-built Prometheus is costly for small- and medium-sized enterprises (SMEs) and insufficient for large enterprises, many are turning to hosted Prometheus.

Solutions

AOM fully interconnects with the open-source Prometheus ecosystem. With Kubernetes clusters connected to Prometheus, enterprises can monitor performance metrics of hosts and Kubernetes clusters through Grafana dashboards.

- Collect metrics through kube-prometheus-stack, self-built Kubernetes clusters, ServiceMonitor, and PodMonitor to monitor service data deployed in CCE clusters.
- Various alarm templates help you quickly detect and locate faults.

4 Comparison Between AOM 1.0 and AOM 2.0

Based on AOM 1.0 functions and common application monitoring, AOM 2.0 collects and monitors more metrics and log data, and displays monitoring results in a visualized manner.

This section compares AOM 1.0 with AOM 2.0.

Table 4-1 Comparison between AOM 1.0 and AOM 2.0

Function		Description	AOM 1.0	AOM 2.0
Resource monitoring	Access center	Quickly connect metrics at the business, application, middleware, and infrastructure layers for monitoring.	Not supported.	Supported.
	Dashboard	Resource metrics, logs, and performance data are displayed in multiple graphs on the same screen.	Partially supported. Only metric data and system performance data can be monitored in a visualized manner.	Supported.
	Alarm management	You can set event conditions for services or set threshold criteria for resource metrics. When an alarm is generated due to an exception in AOM or a related service, the alarm information is sent to the specified personnel by email, SMS, or WeCom.	Partially supported. During alarm rule creation, metrics can be selected by metric type or running Prometheus commands, but cannot be selected from full metrics.	Supported.

Function		Description	AOM 1.0	AOM 2.0
	Container insights	AOM monitors CCE resource usage, status, and alarms from workload and cluster dimensions for fast response and smooth workload running.	Supported.	Supported.
	Metric browsing	You can monitor metric data and trends of each resource and log data in real time, and create alarm rules for metrics to view services and analyze associated data in real time.	Partially supported. Only metric data can be monitored and analyzed.	Supported.
	Infrastructure monitoring	The running status of hosts and cloud services, and VM CPU, memory, and disk information can be monitored in real time.	Supported.	Supported.
	Prometheus monitoring	AOM is fully interconnected with the open-source Prometheus ecosystem, monitors various components, provides multiple preset monitoring dashboards for out-of-the-box availability, and flexibly expands cloud native component metric plug-ins.	Not supported.	Supported.
	Business monitoring	ELB log data reported to LTS are extracted as metrics for unified management. This facilitates real-time monitoring on the metric browsing and dashboard pages.	Not supported.	Supported.
	Log analysis	You can quickly search for required logs from massive quantities of logs. You can also quickly locate faults by analyzing the log source and context.	Supported.	Supported.

Function		Description	AOM 1.0	AOM 2.0
	Process monitoring	Rules can be set to discover deployed applications and collect associated metrics. Drill-down (from applications to components, instances, and containers) is also supported. Applications and components can be monitored from multiple dimensions.	Supported.	Supported.
	Collection management	You can use UniAgents to schedule collection tasks to collect data. UniAgents can be installed manually or automatically.	Not supported.	Supported.

As functions of AOM 1.0 are gradually replaced by those of AOM 2.0, AOM 1.0 will be brought offline soon. You are advised to upgrade AOM 1.0 to AOM 2.0. For details, see **Upgrading to AOM 2.0**.

5 Relationships Between AOM and Other Services

AOM can work with Simple Message Notification (SMN), Distributed Message Service (DMS), and Cloud Trace Service (CTS). For example, when you subscribe to SMN, AOM can inform related personnel of alarm rule status changes by email or Short Message Service (SMS) message. When AOM interconnects with middleware services such as Virtual Private Cloud (VPC) and Elastic Load Balance (ELB), you can monitor them in AOM. When AOM interconnects with Cloud Container Engine (CCE) or Cloud Container Instance (CCI), you can monitor their basic resources and applications, and view related logs and alarms.

SMN

SMN can push notifications based on requirements, and you can receive notifications by SMS message, email, or app. You can also integrate application functions through SMN to reduce system complexity.

AOM uses the message transmission mechanism of SMN. When it is inconvenient for you to query threshold rule status changes on site, AOM sends such changes to you by email or SMS messages. In this way, you can obtain resource status and other information in real time and take necessary measures to avoid service loss.

OBS

Object Storage Service (OBS) is a secure, reliable, and cost-effective cloud storage service. With OBS, you can easily create, modify, and delete buckets, as well as upload, download, and delete objects.

AOM allows you to dump logs to OBS buckets for long-term storage.

LTS

Log Tank Service (LTS) can collect, analyze, and store log data. You can use LTS for efficient device O&M, service trend analysis, security audits, and monitoring.

AOM is a unified entry for Huawei Cloud observability analysis. It does not provide log functions, but integrates them from LTS.

CTS

CTS records operations on cloud resources in your account. Based on the records, you can perform security analysis, trace resource changes, conduct compliance audits, and locate faults. To store operation records for a longer time, you can subscribe to OBS and synchronize operation records to OBS in real time.

With CTS, you can record operations associated with AOM for future query, audit, and tracing.

IAM

Identity and Access Management (IAM) provides identity authentication, permission management, and access control.

IAM can implement authentication and fine-grained authorization for AOM.

Cloud Eye

Cloud Eye provides a multi-dimensional monitoring platform for resources such as Elastic Cloud Server (ECS) and bandwidth. With Cloud Eye, you can view the resource usage and service running status in the cloud, and respond to exceptions in a timely manner to ensure smooth running of services.

VPC

VPC is a logically isolated virtual network. It is created for ECS servers, and supports custom configuration and management, improving resource security and simplifying network deployment.

After subscribing to VPC, you can monitor VPC running status and metrics on the AOM console without installing other plug-ins.

ELB

ELB distributes access traffic to multiple backend ECS servers based on forwarding policies. By distributing traffic, ELB expands the capabilities of application systems to provide services externally. By preventing single points of failures, ELB improves the availability of application systems.

After subscribing to ELB, you can monitor ELB running status and metrics on the AOM console without installing other plug-ins.

RDS

RDS is a cloud-based web service which is reliable, scalable, easy to manage, and ready to use out-of-the-box.

After subscribing to RDS, you can monitor RDS running status and metrics on the AOM console without installing other plug-ins.

DCS

DCS is an online, distributed, in-memory cache service compatible with Redis, Memcached, and In-Memory Data Grid (IMDG). It is reliable, scalable, ready to

use out-of-the-box, and easy to manage, meeting your requirements for high read/write performance and fast data access.

After subscribing to DCS, you can monitor DCS running status and metrics on the AOM console without installing other plug-ins.

CCE

CCE is a high-performance and scalable container service through which enterprises can build reliable containerized applications. It integrates network and storage capabilities, and is compatible with Kubernetes and Docker container ecosystems. CCE enables you to create and manage diverse containerized workloads easily. It also provides efficient O&M capabilities, such as container fault self-healing, monitoring log collection, and auto scaling.

You can monitor basic resources, applications, logs, and alarms about CCE on the AOM console.

ServiceStage

ServiceStage is a one-stop PaaS service that provides cloud-based application hosting, simplifying application lifecycle management, from deployment, monitoring, O&M, to governance. It provides a microservice framework compatible with mainstream open-source ecosystems and enables quick building of distributed applications.

You can monitor basic resources, applications, logs, and alarms about ServiceStage on the AOM console.

FunctionGraph

FunctionGraph hosts and computes functions in a serverless context. It automatically scales up/down resources during peaks and spikes without requiring the reservation of dedicated servers or capacities. Resources are billed on a pay-per-use basis.

You can monitor basic resources, applications, logs, and alarms about FunctionGraph on the AOM console.

ECS

An ECS is a computing server consisting of CPU, memory, image, and Elastic Volume Service (EVS) disk. It supports on-demand allocation and auto scaling. ECSs integrate VPC, virtual firewall, and multi-data-copy capabilities to create an efficient, reliable, and secure computing environment. This ensures stable and uninterrupted running of services. After creating an ECS server, you can use it like using your local computer or physical server.

When purchasing an ECS, ensure that its OS meets the requirements in **Table 7-2**. In addition, install a UniAgent on the ECS. Otherwise, the ECS cannot be monitored by AOM. You can monitor basic resources, applications, logs, and alarms about this ECS on the AOM console.

BMS

A Bare Metal Server (BMS) is a dedicated physical server in the cloud. It provides high-performance computing and ensures data security for core databases, key application systems, and big data. With the advantage of scalable cloud resources, you can apply for BMS servers flexibly and they are billed on a pay-per-use basis.

When purchasing a BMS server, ensure that its OS meets the requirements in **Table 7-2**. In addition, install a UniAgent on the server. Otherwise, the server cannot be monitored by AOM. You can monitor basic resources, applications, logs, and alarms about this server on the AOM console.

6 Comparison Between AOM 2.0 and Cloud Eye

This section compares the cloud service monitoring functions of AOM 2.0 and Cloud Eye.

AOM metric data comes from Cloud Eye. AOM's metric data is in Prometheus format while Cloud Eye's metric data is in a custom format. **Table 6-1** compares the cloud service monitoring functions of AOM and Cloud Eye.

Table 6-1 Comparing the cloud service monitoring functions of AOM and Cloud Eye

Function	Cloud Eye	AOM 2.0
Customization of data storage duration	Not supported (default: 3 months).	Supported (up to 367 days).
Data export	• Aggregated data of the last three months can be exported. • Raw data of the last 48 hours can be exported.	• Dashboards and APIs can be exported.
Aggregate query	Only simple query is supported.	Multi-instance aggregation query is supported. For example, aggregation by tag or resource group.
PromQL syntax	Not supported.	Supported when you use alarm rules and dashboards, and browse metrics.
Dashboards	Single-instance dashboards are supported for standard cloud products.	Various preset templates are provided.

Function	Cloud Eye	AOM 2.0
Graph types supported by dashboards	2	8+
Monitoring views supported by a dashboard	50	100+
Alarm rules that can be created	Max.: 1000.	Default: 3000+. More than 10,000 rules can be supported.
Alarm rules that can be added to an alarm template	Max.: 50.	More than 20 cloud services can be added, and more than 100 alarm rules can be added for each cloud service.
Time that the alarm history can be kept	7 days.	1 year.
Objects that can be selected for single alarm rule creation	5000	Not limited. You can select all resources, and implement regular expression or exact match.
Alarm aggregation	Not supported.	Alarm aggregation based on PromQL syntax is supported.
Connecting to the on-premises Grafana	Not supported.	Prometheus data sources can be directly connected to on-premises Grafana.
Interconnecting with on-premises self-built Prometheus	Not supported.	Data can be directly written to self-built Prometheus.
Business monitoring	Not supported.	Business monitoring based on Prometheus, LTS logs, and custom channels is supported.
On-premises IDC monitoring	Not supported.	Prometheus Exporter-based on-premises hardware, storage, and network monitoring is supported.

Function	Cloud Eye	AOM 2.0
On-premises middleware monitoring	Not supported.	On-premises middleware such as MongoDB, Redis, and RocketMQ can be monitored.

7 Restrictions

Resource Monitoring Restrictions

Table 7-1 Resource monitoring restrictions

Category	Object	Restriction
Dashboard	Dashboard	A maximum of 1000 dashboards can be created in a region.
	Graph	A maximum of 30 graphs can be added to a dashboard.
	Resources, threshold rules, components, or hosts in a graph	• A maximum of 12 resources can be added to a digit graph. Only one resource can be displayed. By default, the first resource is displayed. • A maximum of ten threshold rules can be added to a threshold status graph. • A maximum of ten hosts can be added to a host status graph. • A maximum of ten components can be added to a component status graph.
Metric	Metric data	• Basic edition: Metric data can be stored for up to 7 days. • Professional edition: Metric data can be stored for up to 30 days.
	Metric item	After resources (such as clusters, components, and hosts) are deleted, their metric items can still be stored for up to 30 days.
	Dimension	A maximum of 20 dimensions can be configured for a metric.
	Metric query API	A maximum of 20 metrics can be queried at a time.

Category	Object	Restriction
	Statistical period	The maximum statistical period is 1 hour.
	Data points returned for a single query	A maximum of 1440 data points can be returned each time.
	Custom metric	No restrictions.
	Custom metric reported	A single request cannot exceed 40 KB. The timestamp of a reported metric cannot be 10 minutes later than the standard UTC time. In addition, out-of-order metrics are not received. That is, if a metric is reported at a certain time point, the metrics of earlier time points cannot be reported.
	Application metric Job metric	• When the number of containers on a host exceeds 1000, the ICAgent stops collecting application metrics and sends the **ICAgent Stopped Collecting Application Metrics** alarm (ID: 34105). • When the number of containers on a host within 1000, the ICAgent resumes the collection of application metrics and the **ICAgent Stopped Collecting Application Metrics** alarm is cleared. A job automatically exits after it is completed. To monitor metrics of a job, ensure that its survival time is greater than 90s so that the ICAgent can collect its metric data.
	Resources consumed by the ICAgent	When the ICAgent collects basic metrics, the resources consumed by the ICAgent are related to the number of containers and processes. On a VM without any services, the ICAgent consumes 30 MB memory and records 1% CPU usage. To ensure collection reliability, ensure that fewer than 1000 containers run on a single node.
Alarm rule	Alarm rule	A maximum of 3,000 alarm rules (including metric alarm rules and event alarm rules) can be created.
	Alarm template	A maximum of 150 alarm templates can be created.
Log	Restrictions on the log function	For more information, see **LTS Usage Restrictions**.

Category	Object	Restriction
	Log file	Only text log files can be collected. Other types of log files (for example, binary files) cannot be collected.
		The ICAgent can collect a maximum of 20 log files from a volume mounting directory.
		The ICAgent can collect a maximum of 1000 standard container output log files. These files must be in JSON format.
	Resources consumed during log file collection	The resources consumed during log file collection are closely related to the log volume, number of files, network bandwidth, and backend service processing capability.
	Log discarding	When a single log line exceeds 10,240 bytes, it will be discarded.

Category	Object	Restriction
	Log collection path	**Linux** ● Collection paths support recursion. You can use double asterisks (**) to collect logs from up to five directory levels. Example: **/var/logs/**/a.log** ● Collection paths support fuzzy match. You can use an asterisk (*) to represent one or more characters of a directory or file name. Example: **/var/logs/*/a.log** or **/var/logs/service/a*.log** ● If the collection path is set to a directory, for example, **/var/logs/**, only **.log**, **.trace**, and **.out** files in the directory are collected. If the collection path is set to name of a text file, that file is directly collected. ● Each collection path must be unique. That is, the same path of the same host cannot be configured for different log groups and log streams. **Windows** ● Collection paths support recursion. You can use double asterisks (**) to collect logs from up to five directory levels. Example: **C:\var\service**\a.log** ● Collection paths support fuzzy match. You can use an asterisk (*) to represent one or more characters of a directory or file name. Examples: **C:\var\service*\a.log** and **C:\var\service\a*.log** ● Each collection path must be unique. That is, the same path of the same host cannot be configured for different log groups and log streams. ● Each collection path must be unique. That is, the same path of the same host cannot be configured for different log groups and log streams.
	Log repetition	When the ICAgent is restarted, identical data may be collected around the restart time.
	Historical logs	The storage duration and prices of log data vary according to editions.
Alarm list	Alarms	You can query alarms generated within 31 days in the last year.
	Events	You can query events generated within 31 days in the last year.

Category	Object	Restriction
Application discovery	Application discovery rules	A maximum of 100 application discovery rules can be created.

Collection Management Restrictions

- OS Restrictions

Table 7-2 Linux OSs and versions supported by UniAgent

OS	Version				
Euler OS	1.1 64-bit	2.0 64-bit			
Cent OS	7.1 64-bit	7.2 64-bit	7.3 64-bit	7.4 64-bit	7.5 64-bit
	7.6 64-bit	7.7 64-bit	7.8 64-bit	7.9 64-bit	8.0 64-bit
Ubun tu	16.04 server 64-bit	18.04 server 64-bit	20.04 server 64-bit	22.04 server 64-bit	

NOTE

- For Linux x86_64 hosts, all the OSs and versions listed in the preceding table are supported.
- For Linux Arm hosts, CentOS 7.4/7.5/7.6, EulerOS 2.0, and Ubuntu 18.04 are supported.

Table 7-3 Windows OSs and versions supported by UniAgent

OS	Version
Windows Server	Windows Server 2012 R2 Standard 64-bit
	Windows Server 2012 R2 Standard English 64-bit
	Windows Server 2012 R2 Datacenter 64-bit
	Windows Server 2012 R2 Datacenter English 64-bit
	Windows Server 2016 Standard 64-bit
	Windows Server 2016 Standard English 64-bit
	Windows Server 2016 Datacenter 64-bit

OS	Version
	Windows Server 2016 Datacenter English 64-bit
	Windows Server 2019 Standard 64-bit
	Windows Server 2019 Standard English 64-bit
	Windows Server 2019 Datacenter 64-bit
	Windows Server 2019 Datacenter English 64-bit

- Resource Restrictions

  Table 7-4 Resource restrictions

Object	Restriction
Agent client	When the average CPU usage is greater than 50% or the memory is greater than 100 MB for two minutes, the Agent client automatically restarts.
Agent installation, upgrade, or uninstallation	You can install, upgrade, or uninstall Agents for a maximum of 100 hosts at a time.
Host deletion	You can delete a maximum of 50 hosts with Agents uninstalled at a time.

8 Metric Overview

8.1 Introduction

Metrics reflect resource performance data or status. A metric consists of a **namespace**, **dimension**, name, and unit.

Metric Namespaces

A namespace is an abstract collection of resources and objects. Metrics in different namespaces are independent of each other so that metrics of different applications will not be aggregated to the same statistics information.

- Namespaces of system metrics are fixed and started with **PAAS.**. For details, see **Table 8-1**.

  Table 8-1 Namespaces of system metrics

Namespace	Description
PAAS.AGGR	Namespace of cluster metrics
PAAS.NODE	Namespace of host, network, disk, and file system metrics
PAAS.CONTAINER	Namespace of component, instance, process, and container metrics
PAAS.SLA	Namespace of SLA metrics

- Namespaces of custom metrics must be in the XX.XX format. Each namespace must be 3 to 32 characters long, starting with a letter (excluding **PAAS.**, **SYS.**, and **SRE.**). Only digits, letters, and underscores (_) are allowed.

Metric Dimensions

Metric dimensions indicate the categories of metrics. Each metric has certain features, and a dimension may be considered as a category of such features.

- Dimensions of system metrics are fixed. Different types of metrics have different dimensions. For details, see **8.8 Metric Dimensions**.

- Dimensions of custom metrics must be 1 to 32 characters long, which need to be customized.

8.2 Basic Metrics: VM Metrics

This section describes the categories, names, and meanings of VM metrics reported by ICAgents to AOM.

Table 8-2 VM metrics

Category	Metric	Metric Name	Description	Value Range	Unit
Network metrics	aom_node_network_receive_bytes	Downlink Rate (BPS)	Inbound traffic rate of a measured object	≥ 0	Bytes/s
	aom_node_network_receive_packets	Downlink Rate (PPS)	Number of data packets received by a NIC per second	≥ 0	Packets/s
	aom_node_network_receive_error_packets	Downlink Error Rate	Number of error packets received by a NIC per second	≥ 0	Count/s
	aom_node_network_transmit_bytes	Uplink Rate (BPS)	Outbound traffic rate of a measured object	≥ 0	Bytes/s
	aom_node_network_transmit_error_packets	Uplink Error Rate	Number of error packets sent by a NIC per second	≥ 0	Count/s
	aom_node_network_transmit_packets	Uplink Rate (PPS)	Number of data packets sent by a NIC per second	≥ 0	Packets/s
	aom_node_network_total_bytes	Total Rate (BPS)	Total inbound and outbound traffic rate of a measured object	≥ 0	Bytes/s
Disk metrics	aom_node_disk_read_kilobytes	Disk Read Rate	Volume of data read from a disk per second	≥ 0	KB/s
	aom_node_disk_write_kilobytes	Disk Write Rate	Volume of data written into a disk per second	≥ 0	KB/s

Category	Metric	Metric Name	Description	Value Range	Unit
Disk partition metrics	aom_host_diskpartition_thinpool_metadata_percent	Thin Pool's Metadata Space Usage	Percentage of the thin pool's used metadata space to the total metadata space on a CCE node	0–100	%
	aom_host_diskpartition_thinpool_data_percent	Thin Pool's Data Space Usage	Percentage of the thin pool's used data space to the total data space on a CCE node	0–100	%
	aom_host_diskpartition_total_capacity_megabytes	Thin Pool's Disk Partition Space	Total thin pool's disk partition space on a CCE node	≥ 0	MB
File system metrics	aom_node_disk_available_capacity_megabytes	Available Disk Space	Disk space that has not been used	≥ 0	MB
	aom_node_disk_capacity_megabytes	Total Disk Space	Total disk space	≥ 0	MB
	aom_node_disk_rw_status	Disk Read/ Write Status	Read or write status of a disk	0 or 1 • 0: read / write • 1: read -only	N/A
	aom_node_disk_usage	Disk Usage	Percentage of the used disk space to the total disk space	0–100	%
Host metrics	aom_node_cpu_limit_core	Total CPU Cores	Total number of CPU cores that have been applied for a measured object	≥ 1	Cores
	aom_node_cpu_used_core	Used CPU Cores	Number of CPU cores used by a measured object	≥ 0	Cores

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_node_cpu_usage	CPU Usage	CPU usage of a measured object	0–100	%
	aom_node_memory_free_megabytes	Available Physical Memory	Available physical memory of a measured object	≥ 0	MB
	aom_node_virtual_memory_free_megabytes	Available Virtual Memory	Available virtual memory of a measured object	≥ 0	MB
	aom_node_gpu_memory_free_megabytes	GPU Memory Capacity	Total GPU memory of a measured object	> 0	MB
	aom_node_gpu_memory_usage	GPU Memory Usage	Percentage of the used GPU memory to the total GPU memory	0–100	%
	aom_node_gpu_memory_used_megabytes	Used GPU Memory	GPU memory used by a measured object	≥ 0	MB
	aom_node_gpu_usage	GPU Usage	GPU usage of a measured object	0–100	%
	aom_node_npu_memory_free_megabytes	Total NPU Memory	Total NPU memory of a measured object **NOTE** Only NPU metrics of CCE hosts can be collected.	> 0	MB
	aom_node_npu_memory_usage	NPU Memory Usage	Percentage of the used NPU memory to the total NPU memory **NOTE** Only NPU metrics of CCE hosts can be collected.	0–100	%
	aom_node_npu_memory_used_megabytes	Used NPU Memory	NPU memory used by a measured object **NOTE** Only NPU metrics of CCE hosts can be collected.	≥ 0	MB

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_node_npu_usage	NPU Usage	NPU usage of a measured object **NOTE** Only NPU metrics of CCE hosts can be collected.	0–100	%
	aom_node_npu_temperature_centigrade	NPU Temperature	NPU temperature of a measured object **NOTE** Only NPU metrics of CCE hosts can be collected.	-	°C
	aom_node_memory_usage	Physical Memory Usage	Percentage of the used physical memory to the total physical memory applied for a measured object	0–100	%
	aom_node_status	Host Status	Host status	• 0: Normal • 1: Abnormal	N/A
	aom_node_ntp_offset_ms	NTP Offset	Offset between the local time of the host and the NTP server time. The closer the NTP offset is to 0, the closer the local time of the host is to the time of the NTP server.	-	ms

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_node_ntp_server_status	NTP Server Status	Whether the host is connected to the NTP server	0 or 1 • 0: Connected • 1: Not connected	N/A
	aom_node_ntp_status	NTP Synchronization Status	Whether the local time of the host is synchronized with the NTP server time	0 or 1 • 0: Synchronous • 1: Asynchronous	N/A
	aom_node_process_number	Processes	Number of processes on a measured object	≥ 0	N/A
	aom_node_gpu_temperature_centigrade	GPU Temperature	GPU temperature of a measured object	-	°C
	aom_node_memory_total_megabytes	Total Physical Memory	Total physical memory that has been applied for a measured object	≥ 0	MB
	aom_node_virtual_memory_total_megabytes	Virtual Memory Size	Total virtual memory of a measured object	≥ 0	MB
	aom_node_virtual_memory_usage	Virtual Memory Usage	Percentage of the used virtual memory to the total virtual memory	0–100	%
	aom_node_current_threads_num	Current Threads	Number of threads created on a host	≥ 0	N/A

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_node_sys_max_threads_num	Max Threads	Maximum number of threads that can be created on a host	≥ 0	N/A
	aom_node_phy_disk_total_capacity_megabytes	Total Physical Disk Space	Total disk space of a host	≥ 0	MB
	aom_node_physical_disk_total_used_megabytes	Used Physical Disk Space	Used disk space of a host	≥ 0	MB
	aom_billing_hostUsed	Hosts	Number of hosts connected per day	≥ 0	N/A
Cluster metrics	aom_cluster_cpu_limit_core	Total CPU Cores	Total number of CPU cores that have been applied for a measured object	≥ 1	Cores
	aom_cluster_cpu_used_core	Used CPU Cores	Number of CPU cores used by a measured object	≥ 0	Cores
	aom_cluster_cpu_usage	CPU Usage	CPU usage of a measured object	0–100	%
	aom_cluster_disk_available_capacity_megabytes	Available Disk Space	Disk space that has not been used	≥ 0	MB
	aom_cluster_disk_capacity_megabytes	Total Disk Space	Total disk space	≥ 0	MB
	aom_cluster_disk_usage	Disk Usage	Percentage of the used disk space to the total disk space	0–100	%
	aom_cluster_memory_free_megabytes	Available Physical Memory	Available physical memory of a measured object	≥ 0	MB

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_cluster_virtual_memory_free_megabytes	Available Virtual Memory	Available virtual memory of a measured object	≥ 0	MB
	aom_cluster_gpu_memory_free_megabytes	Available GPU Memory	Available GPU memory of a measured object	> 0	MB
	aom_cluster_gpu_memory_usage	GPU Memory Usage	Percentage of the used GPU memory to the total GPU memory	0–100	%
	aom_cluster_gpu_memory_used_megabytes	Used GPU Memory	GPU memory used by a measured object	≥ 0	MB
	aom_cluster_gpu_usage	GPU Usage	GPU usage of a measured object	0–100	%
	aom_cluster_memory_usage	Physical Memory Usage	Percentage of the used physical memory to the total physical memory applied for a measured object	0–100	%
	aom_cluster_network_receive_bytes	Downlink Rate (BPS)	Inbound traffic rate of a measured object	≥ 0	Bytes/s
	aom_cluster_network_transmit_bytes	Uplink Rate (BPS)	Outbound traffic rate of a measured object	≥ 0	Bytes/s
	aom_cluster_memory_total_megabytes	Total Physical Memory	Total physical memory that has been applied for a measured object	≥ 0	MB
	aom_cluster_virtual_memory_total_megabytes	Virtual Memory Size	Total virtual memory of a measured object	≥ 0	MB
	aom_cluster_virtual_memory_usage	Virtual Memory Usage	Percentage of the used virtual memory to the total virtual memory	0–100	%

Category	Metric	Metric Name	Description	Value Range	Unit
Container metrics	aom_container_cpu_limit_core	Total CPU Cores	Total number of CPU cores restricted for a measured object	≥ 1	Cores
	aom_container_cpu_used_core	Used CPU Cores	Number of CPU cores used by a measured object	≥ 0	Cores
	aom_container_cpu_usage	CPU Usage	CPU usage of a measured object Percentage of the used CPU cores to the total CPU cores restricted for a measured object	0–100	%
	aom_container_disk_read_kilobytes	Disk Read Rate	Volume of data read from a disk per second	≥ 0	KB/s
	aom_container_disk_write_kilobytes	Disk Write Rate	Volume of data written into a disk per second	≥ 0	KB/s
	aom_container_filesystem_available_capacity_megabytes	Available File System Capacity	Available file system capacity of a measured object. This metric is available only for containers using the Device Mapper storage drive in the Kubernetes cluster of version 1.11 or later.	≥ 0	MB

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_container_filesystem_capacity_megabytes	Total File System Capacity	Total file system capacity of a measured object. This metric is available only for containers using the Device Mapper storage drive in the Kubernetes cluster of version 1.11 or later.	≥ 0	MB
	aom_container_filesystem_usage	File System Usage	File system usage of a measured object. That is, the percentage of the used file system to the total file system. This metric is available only for containers using the Device Mapper storage drive in the Kubernetes cluster of version 1.11 or later.	0–100	%
	aom_container_gpu_memory_free_megabytes	GPU Memory Capacity	Total GPU memory of a measured object	> 0	MB
	aom_container_gpu_memory_usage	GPU Memory Usage	Percentage of the used GPU memory to the total GPU memory	0–100	%
	aom_container_gpu_memory_used_megabytes	Used GPU Memory	GPU memory used by a measured object	≥ 0	MB
	aom_container_gpu_usage	GPU Usage	GPU usage of a measured object	0–100	%
	aom_container_npu_memory_free_megabytes	Total NPU Memory	Total NPU memory of a measured object	> 0	MB

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_container_npu_memory_usage	NPU Memory Usage	Percentage of the used NPU memory to the total NPU memory	0–100	%
	aom_container_npu_memory_used_megabytes	Used NPU Memory	NPU memory used by a measured object	≥ 0	MB
	aom_container_npu_usage	NPU Usage	NPU usage of a measured object	0–100	%
	aom_container_memory_request_megabytes	Total Physical Memory	Total physical memory restricted for a measured object	≥ 0	MB
	aom_container_memory_usage	Physical Memory Usage	Percentage of the used physical memory to the total physical memory restricted for a measured object	0–100	%
	aom_container_memory_used_megabytes	Used Physical Memory	Used physical memory of a measured object	≥ 0	MB
	aom_container_network_receive_bytes	Downlink Rate (BPS)	Inbound traffic rate of a measured object	≥ 0	Bytes/s
	aom_container_network_receive_packets	Downlink Rate (PPS)	Number of data packets received by a NIC per second	≥ 0	Packets/s
	aom_container_network_receive_error_packets	Downlink Error Rate	Number of error packets received by a NIC per second	≥ 0	Count/s
	aom_container_network_rx_error_packets	Error Packets Received	Number of error packets received by a measured object	≥ 0	Count
	aom_container_network_transmit_bytes	Uplink Rate (BPS)	Outbound traffic rate of a measured object	≥ 0	Bytes/s

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_container_network_transmit_error_packets	Uplink Error Rate	Number of error packets sent by a NIC per second	≥ 0	Count/s
	aom_container_network_transmit_packets	Uplink Rate (PPS)	Number of data packets sent by a NIC per second	≥ 0	Packets/s
	aom_process_status	Status	Docker container status	0 or 1 • 0: Normal • 1: Abnormal	N/A
	aom_container_memory_workingset_usage	Working Set Memory Usage	Usage of the working set memory	0–100	%
	aom_container_memory_workingset_used_megabytes	Used Working Set Memory	Working set memory that has been used	≥ 0	MB
Process metrics	aom_process_cpu_limit_core	Total CPU Cores	Total number of CPU cores that have been applied for a measured object	≥ 1	Cores
	aom_process_cpu_used_core	Used CPU Cores	Number of CPU cores used by a measured object	≥ 0	Cores
	aom_process_cpu_usage	CPU Usage	CPU usage of a measured object Percentage of the used CPU cores to the CPU cores that have been applied	0–100	%
	aom_process_handle_count	Handles	Number of handles used by a measured object	≥ 0	N/A

Category	Metric	Metric Name	Description	Value Range	Unit
	aom_process_max_handle_count	Max Handles	Maximum number of handles used by a measured object	≥ 0	N/A
	aom_process_memory_request_megabytes	Total Physical Memory	Total physical memory that has been applied for a measured object	≥ 0	MB
	aom_process_memory_usage	Physical Memory Usage	Percentage of the used physical memory to the total physical memory applied for a measured object	0–100	%
	aom_process_memory_used_megabytes	Used Physical Memory	Used physical memory of a measured object	≥ 0	MB
	aom_process_status	Status	Process status	0 or 1 • 0: Normal • 1: Abnormal	N/A
	aom_process_thread_count	Threads	Number of threads used by a measured object	≥ 0	N/A
	aom_process_virtual_memory_total_megabytes	Virtual Memory Size	Total virtual memory that has been applied for a measured object	≥ 0	MB

NOTE

- If the host type is **CCE**, you can view disk partition metrics. The supported OSs are CentOS 7.6 and EulerOS 2.5.
- Log in to the CCE node as the **root** user and run the **docker info | grep 'Storage Driver'** command to check the Docker storage driver type. If the command output shows driver type **Device Mapper**, the thin pool metrics can be viewed. Otherwise, the thin pool metrics cannot be viewed.
- Memory usage = (Physical memory capacity – Available physical memory capacity)/ Physical memory capacity; Virtual memory usage = ((Physical memory capacity + Total virtual memory capacity) – (Available physical memory capacity + Available virtual memory capacity))/(Physical memory capacity + Total virtual memory capacity) Currently, the virtual memory of a newly created VM is 0 MB by default. If no virtual memory is configured, the memory usage on the monitoring page is the same as the virtual memory usage.
- For the total and used physical disk space, only the space of the local disk partitions' file systems is counted. The file systems (such as JuiceFS, NFS, and SMB) mounted to the host through the network are not taken into account.
- Cluster metrics are aggregated by AOM based on host metrics, and do not include the metrics of master hosts.

8.3 Basic Metrics: Container Metrics

This section describes the categories, names, and meanings of metrics reported to AOM from CCE's kube-prometheus-stack add-on or on-premises Kubernetes clusters.

Table 8-3 Metrics of containers running in CCE or on-premises Kubernetes clusters

Target Name	Job Name	Metric	Description
• serviceMonitor/monitoring/coredns/0 • serviceMonitor/monitoring/node-local-dns/0	coredns and node-local-dns	coredns_build_info	Information to build CoreDNS
		coredns_cache_entries	Number of entries in the cache
		coredns_cache_size	Cache size
		coredns_cache_hits_total	Number of cache hits
		coredns_cache_misses_total	Number of cache misses
		coredns_cache_requests_total	Total number of DNS resolution requests in different dimensions
		coredns_dns_request_duration_seconds_bucket	Histogram of DNS request duration (bucket)

Target Name	Job Name	Metric	Description
		coredns_dns_request_duration_seconds_count	Histogram of DNS request duration (count)
		coredns_dns_request_duration_seconds_sum	Histogram of DNS request duration (sum)
		coredns_dns_request_size_bytes_bucket	Histogram of the size of DNS request (bucket)
		coredns_dns_request_size_bytes_count	Histogram of the size of DNS request (count)
		coredns_dns_request_size_bytes_sum	Histogram of the size of DNS request (sum)
		coredns_dns_requests_total	Number of DNS requests
		coredns_dns_response_size_bytes_bucket	Histogram of the size of DNS response (bucket)
		coredns_dns_response_size_bytes_count	Histogram of the size of DNS response (count)
		coredns_dns_response_size_bytes_sum	Histogram of the size of DNS response (sum)
		coredns_dns_responses_total	DNS response codes and number of DNS response codes
		coredns_forward_conn_cache_hits_total	Number of cache hits for each protocol and data flow
		coredns_forward_conn_cache_misses_total	Number of cache misses for each protocol and data flow
		coredns_forward_healthcheck_broken_total	Unhealthy upstream count
		coredns_forward_healthcheck_failures_total	Count of failed health checks per upstream

Target Name	Job Name	Metric	Description
		coredns_forward_max_concurrent_rejects_total	Number of requests rejected due to excessive concurrent requests
		coredns_forward_request_duration_seconds_bucket	Histogram of forward request duration (bucket)
		coredns_forward_request_duration_seconds_count	Histogram of forward request duration (count)
		coredns_forward_request_duration_seconds_sum	Histogram of forward request duration (sum)
		coredns_forward_requests_total	Number of requests for each data flow
		coredns_forward_responses_total	Number of responses to each data flow
		coredns_health_request_duration_seconds_bucket	Histogram of health request duration (bucket)
		coredns_health_request_duration_seconds_count	Histogram of health request duration (count)
		coredns_health_request_duration_seconds_sum	Histogram of health request duration (sum)
		coredns_health_request_failures_total	Number of health request failures
		coredns_hosts_reload_timestamp_seconds	Timestamp of the last reload of the host file
		coredns_kubernetes_dns_programming_duration_seconds_bucket	Histogram of DNS programming duration (bucket)
		coredns_kubernetes_dns_programming_duration_seconds_count	Histogram of DNS programming duration (count)
		coredns_kubernetes_dns_programming_duration_seconds_sum	Histogram of DNS programming duration (sum)
		coredns_local_localhost_requests_total	Number of localhost requests

Target Name	Job Name	Metric	Description
		coredns_nodecache_setup_errors_total	Number of nodecache setup errors
		coredns_dns_response_rcode_count_total	Number of responses for each Zone and Rcode
		coredns_dns_request_count_total	Number of DNS requests
		coredns_dns_request_do_count_total	Number of requests with the DNSSEC OK (DO) bit set
		coredns_dns_do_requests_total	Number of requests with the DO bit set
		coredns_dns_request_type_count_total	Number of requests for each Zone and Type
		coredns_panics_total	Total number of panics
		coredns_plugin_enabled	Whether a plugin is enabled
		coredns_reload_failed_total	Number of last reload failures
serviceMonitor/monitoring/kube-apiserver/0	apiserver	aggregator_unavailable_apiservice	Number of unavailable APIServices
		apiserver_admission_controller_admission_duration_seconds_bucket	Processing delay of an Admission Controller
		apiserver_admission_webhook_admission_duration_seconds_bucket	Processing delay of an Admission Webhook
		apiserver_admission_webhook_admission_duration_seconds_count	Number of Admission Webhook processing requests
		apiserver_client_certificate_expiration_seconds_bucket	Remaining validity period of the client certificate

Target Name	Job Name	Metric	Description
		apiserver_client_certificate_expiration_seconds_count	Remaining validity period of the client certificate
		apiserver_current_inflight_requests	Number of read requests in process
		apiserver_request_duration_seconds_bucket	Delay of the client's access to the APIServer
		apiserver_request_total	Number of different requests to the APIServer
		go_goroutines	Number of goroutines
		kubernetes_build_info	Information to build Kubernetes
		process_cpu_seconds_total	Total process CPU time
		process_resident_memory_bytes	Size of the resident memory set for a process
		rest_client_requests_total	Number of REST requests
		workqueue_adds_total	Number of adds handled by a work queue
		workqueue_depth	Depth of a work queue
		workqueue_queue_duration_seconds_bucket	Duration when a task exists in the work queue
		aggregator_unavailable_apiservice_total	Number of unavailable APIServices
		rest_client_request_duration_seconds_bucket	Histogram of REST request duration
serviceMonitor/monitoring/kubelet/0	kubelet	kubelet_certificate_manager_client_expiration_renew_errors	Number of certificate renewal errors
		kubelet_certificate_manager_client_ttl_seconds	Time-to-live (TTL) of the Kubelet client certificate

Target Name	Job Name	Metric	Description
		kubelet_cgroup_manager_duration_seconds_bucket	Duration of the cgroup manager operations (bucket)
		kubelet_cgroup_manager_duration_seconds_count	Duration of the cgroup manager operations (count)
		kubelet_node_config_error	If a configuration-related error occurs on a node, the value of this metric is **true (1)**. If there is no configuration-related error, the value is **false (0)**.
		kubelet_node_name	Node name. The value is always **1**.
		kubelet_pleg_relist_duration_seconds_bucket	Duration of relisting pods in PLEG (bucket)
		kubelet_pleg_relist_duration_seconds_count	Duration of relisting pods in PLEG (count)
		kubelet_pleg_relist_interval_seconds_bucket	Interval between relisting operations in PLEG (bucket)
		kubelet_pod_start_duration_seconds_count	Time required for starting a single pod (count)
		kubelet_pod_start_duration_seconds_bucket	Time required for starting a single pod (bucket)
		kubelet_pod_worker_duration_seconds_bucket	Duration for synchronizing a single pod. Operation type: create, update, or sync
		kubelet_running_containers	Number of running containers
		kubelet_running_pods	Number of running pods
		kubelet_runtime_operations_duration_seconds_bucket	Duration of the runtime operations (bucket)

Target Name	Job Name	Metric	Description
		kubelet_runtime_operations_errors_total	Number of runtime operation errors listed by operation type
		kubelet_runtime_operations_total	Number of runtime operations listed by operation type
		kubelet_volume_stats_available_bytes	Number of available bytes in a volume
		kubelet_volume_stats_capacity_bytes	Capacity of the volume in bytes
		kubelet_volume_stats_inodes	Total number of inodes in a volume
		kubelet_volume_stats_inodes_used	Number of used inodes in a volume
		kubelet_volume_stats_used_bytes	Number of used bytes in a volume
		storage_operation_duration_seconds_bucket	Duration of each storage operation (bucket)
		storage_operation_duration_seconds_count	Duration of each storage operation (count)
		storage_operation_errors_total	Number of storage operation errors
		volume_manager_total_volumes	Number of volumes in the Volume Manager
		rest_client_requests_total	Number of HTTP client requests partitioned by status code, method, and host
		rest_client_request_duration_seconds_bucket	Request delay (bucket)
		process_resident_memory_bytes	Size of the resident memory set for a process
		process_cpu_seconds_total	Total process CPU time
		go_goroutines	Number of goroutines

Target Name	Job Name	Metric	Description
serviceMonitor/monitoring/kubelet/1	kubelet	container_cpu_cfs_periods_total	Number of elapsed enforcement period intervals
		container_cpu_cfs_throttled_periods_total	Number of throttled period intervals
		container_cpu_cfs_throttled_seconds_total	Total time duration the container has been throttled
		container_cpu_load_average_10s	Value of container CPU load average over the last 10 seconds
		container_cpu_usage_seconds_total	Cumulative CPU time consumed by a container in core-seconds
		container_file_descriptors	Number of open file descriptors for a container
		container_fs_inodes_free	Number of available inodes in a file system
		container_fs_inodes_total	Number of inodes in a file system
		container_fs_io_time_seconds_total	Cumulative seconds spent on doing I/Os by the disk or file system
		container_fs_limit_bytes	Total disk or file system capacity that can be consumed by a container
		container_fs_read_seconds_total	Cumulative number of seconds the container spent on reading disk or file system data
		container_fs_reads_bytes_total	Cumulative amount of disk or file system data read by a container

Target Name	Job Name	Metric	Description
		container_fs_reads_total	Cumulative number of disk or file system reads completed by a container
		container_fs_usage_bytes	File system usage
		container_fs_write_seconds_total	Cumulative number of seconds the container spent on writing data to the disk or file system
		container_fs_writes_bytes_total	Total amount of data written by a container to a disk or file system
		container_fs_writes_total	Cumulative number of disk or file system writes completed by a container
		container_memory_cache	Memory used for the page cache of a container
		container_memory_failcnt	Number of memory usage hits limits
		container_memory_max_usage_bytes	Maximum memory usage recorded for a container
		container_memory_rss	Size of the resident memory set for a container
		container_memory_swap	Container swap usage
		container_memory_usage_bytes	Current memory usage of a container
		container_memory_working_set_bytes	Memory usage of the working set of a container
		container_network_receive_bytes_total	Total volume of data received by the container network

Target Name	Job Name	Metric	Description
		container_network_receive_errors_total	Cumulative number of errors encountered during reception
		container_network_receive_packets_dropped_total	Cumulative number of packets dropped during reception
		container_network_receive_packets_total	Cumulative number of packets received
		container_network_transmit_bytes_total	Total volume of data transmitted on the container network
		container_network_transmit_errors_total	Cumulative number of errors encountered during transmission
		container_network_transmit_packets_dropped_total	Cumulative number of packets dropped during transmission
		container_network_transmit_packets_total	Cumulative number of packets transmitted
		container_spec_cpu_quota	CPU quota of the container
		container_spec_memory_limit_bytes	Memory limit for the container
		machine_cpu_cores	Number of logical CPU cores
		machine_memory_bytes	Amount of memory
serviceMonitor/monitoring/kube-state-metrics/0	kube-state-metrics-prom	kube_cronjob_status_active	Running cronjob
		kube_cronjob_info	Cronjob information
		kube_cronjob_labels	Label of a cronjob
		kube_configmap_info	ConfigMap information
		kube_daemonset_created	DaemonSet creation time
		kube_daemonset_status_current_number_scheduled	Number of DaemonSets that are being scheduled

Target Name	Job Name	Metric	Description
		kube_daemonset_status_desired_number_scheduled	Number of DaemonSets expected to be scheduled
		kube_daemonset_status_number_available	Number of nodes that should be running a DaemonSet pod and have at least one DaemonSet pod running and available
		kube_daemonset_status_number_misscheduled	Number of nodes that are not expected to run a DaemonSet pod
		kube_daemonset_status_number_ready	Number of nodes that should be running the DaemonSet pods and have one or more DaemonSet pods running and ready
		kube_daemonset_status_number_unavailable	Number of nodes that should be running the DaemonSet pods but have none of the DaemonSet pods running and available
		kube_daemonset_status_updated_number_scheduled	Number of nodes that are running an updated DaemonSet pod
		kube_deployment_created	Deployment creation timestamp
		kube_deployment_labels	Deployment labels
		kube_deployment_metadata_generation	Sequence number representing a specific generation of the desired state
		kube_deployment_spec_replicas	Number of desired replicas for a Deployment

Target Name	Job Name	Metric	Description
		kube_deployment_spec_strategy_rollingupdate_max_unavailable	Maximum number of unavailable replicas during a rolling update of a Deployment
		kube_deployment_status_observed_generation	The generation observed by the Deployment controller
		kube_deployment_status_replicas	Number of current replicas of a Deployment
		kube_deployment_status_replicas_available	Number of available replicas per Deployment
		kube_deployment_status_replicas_ready	Number of ready replicas per Deployment
		kube_deployment_status_replicas_unavailable	Number of unavailable replicas per Deployment
		kube_deployment_status_replicas_updated	Number of updated replicas per Deployment
		kube_job_info	Information about the job
		kube_namespace_labels	Namespace labels
		kube_node_labels	Node labels
		kube_node_info	Information about a node
		kube_node_spec_taint	Taint of a node
		kube_node_spec_unschedulable	Whether new pods can be scheduled to a node
		kube_node_status_allocatable	Allocatable resources on a node
		kube_node_status_capacity	Capacity for different resources on a node
		kube_node_status_condition	Condition of a node

Target Name	Job Name	Metric	Description
		kube_node_volcano_oversubscription_status	Node oversubscription status
		kube_persistentvolume_status_phase	Phase of a PV status
		kube_persistentvolumeclaim_status_phase	Phase of a PVC status
		kube_persistentvolume_info	Information about a PV
		kube_persistentvolumeclaim_info	Information about a PVC
		kube_pod_container_info	Information about a container running in the pod
		kube_pod_container_resource_limits	Number of container resource limits
		kube_pod_container_resource_requests	Number of container resource requests
		kube_pod_container_status_last_terminated_reason	Last reason the container was in a terminated state
		kube_pod_container_status_ready	Whether the container's readiness check succeeded
		kube_pod_container_status_restarts_total	Number of container restarts
		kube_pod_container_status_running	Whether the container is running.
		kube_pod_container_status_terminated	Whether the container is terminated
		kube_pod_container_status_terminated_reason	The reason why the container is in a terminated state
		kube_pod_container_status_waiting	Whether the container is waiting
		kube_pod_container_status_waiting_reason	The reason why the container is in the waiting state
		kube_pod_info	Information about a pod

Target Name	Job Name	Metric	Description
		kube_pod_labels	Pod labels
		kube_pod_owner	Information about the pod's owner
		kube_pod_status_phase	Current phase of a pod
		kube_pod_status_ready	Whether the pod is ready
		kube_secret_info	Information about a secret
		kube_statefulset_created	StatefulSet creation timestamp
		kube_statefulset_labels	Information about StatefulSet labels
		kube_statefulset_metadata_generation	Sequence number representing a specific generation of the desired state for a StatefulSet
		kube_statefulset_replicas	Number of desired pods for a StatefulSet
		kube_statefulset_status_observed_generation	The generation observed by the StatefulSet controller
		kube_statefulset_status_replicas	Number of replicas per StatefulSet
		kube_statefulset_status_replicas_ready	Number of ready replicas per StatefulSet
		kube_statefulset_status_replicas_updated	Number of updated replicas per StatefulSet
		kube_job_spec_completions	Desired number of successfully finished pods that should run with the job
		kube_job_status_failed	Failed jobs
		kube_job_status_succeeded	Successful jobs

Target Name	Job Name	Metric	Description
		kube_node_status_allocatable_cpu_cores	Number of allocatable CPU cores of a node
		kube_node_status_allocatable_memory_bytes	Total allocatable memory of a node
		kube_replicaset_owner	Information about the ReplicaSet's owner
		kube_resourcequota	Information about resource quota
		kube_pod_spec_volumes_persistentvolume-claims_info	Information about the PVC associated with the pod
serviceMonitor/monitoring/prometheus-lightweight/0	prometheus-lightweight	vm_persistentqueue_blocks_dropped_total	Number of dropped blocks in a send queue
		vm_persistentqueue_blocks_read_total	Number of blocks read by a send queue
		vm_persistentqueue_blocks_written_total	Number of blocks written to a send queue
		vm_persistentqueue_bytes_pending	Number of pending bytes in a send queue
		vm_persistentqueue_bytes_read_total	Number of bytes read by a send queue
		vm_persistentqueue_bytes_written_total	Number of bytes written to a send queue
		vm_promscrape_active_scrapers	Number of active scrapes
		vm_promscrape_conn_read_errors_total	Number of read errors during scrapes
		vm_promscrape_conn_write_errors_total	Number of write errors during scrapes
		vm_promscrape_max_scrape_size_exceeded_errors_total	Number of failed scrapes due to the exceeded response size
		vm_promscrape_scrape_duration_seconds_sum	Duration of scrapes (sum)

Target Name	Job Name	Metric	Description
		vm_promscrape_scrape_duration_seconds_count	Duration of scrapes (count)
		vm_promscrape_scrapes_total	Number of scrapes
		vmagent_remotewrite_bytes_sent_total	Number of bytes sent via a remote write
		vmagent_remotewrite_duration_seconds_sum	Time required for a remote write (sum)
		vmagent_remotewrite_duration_seconds_count	Time required for a remote write (count)
		vmagent_remotewrite_packets_dropped_total	Number of dropped packets during a remote write
		vmagent_remotewrite_pending_data_bytes	Number of pending bytes during a remote write
		vmagent_remotewrite_requests_total	Number of requests of the remote write
		vmagent_remotewrite_retries_count_total	Number of retries of the remote write
		go_goroutines	Number of goroutines
serviceMonitor/monitoring/node-exporter/0	node-exporter	node_boot_time_seconds	Node boot time
		node_context_switches_total	Number of context switches
		node_cpu_seconds_total	Seconds each CPU spent doing each type of work
		node_disk_io_now	Number of I/Os in progress
		node_disk_io_time_seconds_total	Total seconds spent doing I/Os
		node_disk_io_time_weighted_seconds_total	The weighted number of seconds spent doing I/Os

Target Name	Job Name	Metric	Description
		node_disk_read_bytes_total	Number of bytes that are read
		node_disk_read_time_seconds_total	Number of seconds spent by all reads
		node_disk_reads_completed_total	Number of reads completed
		node_disk_write_time_seconds_total	Number of seconds spent by all writes
		node_disk_writes_completed_total	Number of writes completed
		node_disk_written_bytes_total	Number of bytes that are written
		node_docker_thinpool_data_space_available	Available data space of a docker thin pool
		node_docker_thinpool_metadata_space_available	Available metadata space of a docker thin pool
		node_exporter_build_info	Node exporter build information
		node_filefd_allocated	Allocated file descriptors
		node_filefd_maximum	Maximum number of file descriptors
		node_filesystem_avail_bytes	File system space that is available for use
		node_filesystem_device_error	Whether an error occurred while getting statistics for the given device
		node_filesystem_free_bytes	Remaining space of a file system
		node_filesystem_readonly	Read-only file system
		node_filesystem_size_bytes	Consumed space of a file system
		node_forks_total	Number of forks
		node_intr_total	Number of interruptions that occurred

Target Name	Job Name	Metric	Description
		node_load1	1-minute average CPU load
		node_load15	15-minute average CPU load
		node_load5	5-minute average CPU load
		node_memory_Buffers_bytes	Memory of the node buffer
		node_memory_Cached_bytes	Memory for the node page cache
		node_memory_MemAvailable_bytes	Available memory of a node
		node_memory_MemFree_bytes	Free memory of a node
		node_memory_MemTotal_bytes	Total memory of a node
		node_network_receive_bytes_total	Total amount of received data
		node_network_receive_drop_total	Cumulative number of packets dropped during reception
		node_network_receive_errs_total	Cumulative number of errors encountered during reception
		node_network_receive_packets_total	Cumulative number of packets received
		node_network_transmit_bytes_total	Total amount of transmitted data
		node_network_transmit_drop_total	Cumulative number of dropped packets during transmission
		node_network_transmit_errs_total	Cumulative number of errors encountered during transmission
		node_network_transmit_packets_total	Cumulative number of packets transmitted
		node_procs_blocked	Blocked processes
		node_procs_running	Running processes

Target Name	Job Name	Metric	Description
		node_sockstat_sockets_used	Number of sockets in use
		node_sockstat_TCP_alloc	Number of allocated TCP sockets
		node_sockstat_TCP_inuse	Number of TCP sockets in use
		node_sockstat_TCP_orphan	Number of orphaned TCP sockets
		node_sockstat_TCP_tw	Number of TCP sockets in the **TIME_WAIT** state
		node_sockstat_UDPLITE_inuse	Number of UDP-Lite sockets in use
		node_sockstat_UDP_inuse	Number of UDP sockets in use
		node_sockstat_UDP_mem	UDP socket buffer usage
		node_timex_offset_seconds	Time offset
		node_timex_sync_status	Synchronization status of node clocks
		node_uname_info	Labeled system information as provided by the uname system call
		node_vmstat_oom_kill	OOM kill in **/proc/vmstat**
		process_cpu_seconds_total	Total process CPU time
		process_max_fds	Maximum number of file descriptors of a process
		process_open_fds	Opened file descriptors by a process
		process_resident_memory_bytes	Size of the resident memory set for a process

Target Name	Job Name	Metric	Description
		process_start_time_seconds	Process start time
		process_virtual_memory_bytes	Virtual memory size for a process
		process_virtual_memory_max_bytes	Maximum virtual memory size for a process
		node_netstat_Tcp_ActiveOpens	Number of TCP connections that directly change from the **CLOSED** state to the **SYN-SENT** state
		node_netstat_Tcp_PassiveOpens	Number of TCP connections that directly change from the **LISTEN** state to the **SYN-RCVD** state
		node_netstat_Tcp_CurrEstab	Number of TCP connections in the **ESTABLISHED** or **CLOSE-WAIT** state
		node_vmstat_pgmajfault	Number of major faults per second in **/proc/vmstat**
		node_vmstat_pgpgout	Number of page out between main memory and block device in **/proc/vmstat**
		node_vmstat_pgfault	Number of page faults the system has made per second in **/proc/vmstat**
		node_vmstat_pgpgin	Number of page in between main memory and block device in **/proc/vmstat**
		node_processes_max_processes	PID limit value
		node_processes_pids	Number of PIDs

Target Name	Job Name	Metric	Description
		node_nf_conntrack_entries	Number of currently allocated flow entries for connection tracking
		node_nf_conntrack_entries_limit	Maximum size of a connection tracking table
		promhttp_metric_handler_requests_in_flight	Number of metrics being processed
		go_goroutines	Number of node exporter goroutines
podMonitor/monitoring/nvidia-gpu-device-plugin/0	monitoring/nvidia-gpu-device-plugin	cce_gpu_utilization	GPU compute usage
		cce_gpu_memory_utilization	GPU memory usage
		cce_gpu_encoder_utilization	GPU encoding usage
		cce_gpu_decoder_utilization	GPU decoding usage
		cce_gpu_utilization_process	GPU compute usage of each process
		cce_gpu_memory_utilization_process	GPU memory usage of each process
		cce_gpu_encoder_utilization_process	GPU encoding usage of each process
		cce_gpu_decoder_utilization_process	GPU decoding usage of each process
		cce_gpu_memory_used	Used GPU memory
		cce_gpu_memory_total	Total GPU memory
		cce_gpu_memory_free	Free GPU memory
		cce_gpu_bar1_memory_used	Used GPU BAR1 memory
		cce_gpu_bar1_memory_total	Total GPU BAR1 memory
		cce_gpu_clock	GPU clock frequency
		cce_gpu_memory_clock	GPU memory frequency

Target Name	Job Name	Metric	Description
		cce_gpu_graphics_clock	GPU frequency
		cce_gpu_video_clock	GPU video processor frequency
		cce_gpu_temperature	GPU temperature
		cce_gpu_power_usage	GPU power
		cce_gpu_total_energy_consumption	Total GPU energy consumption
		cce_gpu_pcie_link_bandwidth	GPU PCIe bandwidth
		cce_gpu_nvlink_bandwidth	GPU NVLink bandwidth
		cce_gpu_pcie_throughput_rx	GPU PCIe RX bandwidth
		cce_gpu_pcie_throughput_tx	GPU PCIe TX bandwidth
		cce_gpu_nvlink_utilization_counter_rx	GPU NVLink RX bandwidth
		cce_gpu_nvlink_utilization_counter_tx	GPU NVLink TX bandwidth
		cce_gpu_retired_pages_sbe	Number of GPU single-bit error isolation pages
		cce_gpu_retired_pages_dbe	Number of GPU dual-bit error isolation pages
		xgpu_memory_total	Total xGPU memory
		xgpu_memory_used	Used xGPU memory
		xgpu_core_percentage_total	Total xGPU compute
		xgpu_core_percentage_used	Used xGPU compute

Target Name	Job Name	Metric	Description
		gpu_schedule_policy	There are three GPU modes specified by three values. The value **0** indicates the GPU memory isolation, compute sharing mode. The value **1** indicates the GPU memory and compute isolation mode. The value **2** indicates the default mode, indicating that the GPU is not virtualized.
		xgpu_device_health	Health status of xGPU. The value **0** indicates that the xGPU is healthy, and the value **1** indicates that the xGPU is unhealthy.
serviceMonitor/monitoring/prometheus-server/0	prometheus-server	prometheus_build_info	Information to build Prometheus
		prometheus_engine_query_duration_seconds	Query time
		prometheus_engine_query_duration_seconds_count	Number of queries
		prometheus_sd_discovered_targets	Number of targets discovered by each job
		prometheus_remote_storage_bytes_total	Number of bytes sent
		prometheus_remote_storage_enqueue_retries_total	Number of retries for entering a queue
		prometheus_remote_storage_highest_timestamp_in_seconds	Highest timestamp that has come into the remote storage via the Appender interface, in seconds since epoch

Target Name	Job Name	Metric	Description
		prometheus_remote_storage_queue_highest_sent_timestamp_seconds	Highest timestamp successfully sent by a remote write
		prometheus_remote_storage_samples_dropped_total	Total number of samples read from the WAL but not sent to remote storage
		prometheus_remote_storage_samples_failed_total	Number of samples that failed to be sent to remote storage
		prometheus_remote_storage_samples_in_total	Number of samples read into remote storage
		prometheus_remote_storage_samples_pending	Number of samples pending in shards to be sent to remote storage
		prometheus_remote_storage_samples_retried_total	Number of samples which failed to be sent to remote storage but were retried
		prometheus_remote_storage_samples_total	Total number of samples sent to remote storage
		prometheus_remote_storage_shard_capacity	Capacity of each shard of the queue used for parallel sending to the remote storage
		prometheus_remote_storage_shards	Number of shards used for parallel sending to the remote storage
		prometheus_remote_storage_shards_desired	Number of shards that the queues shard calculation wants to run based on the rate of samples in vs. samples out

Target Name	Job Name	Metric	Description
		prometheus_remote_storage_shards_max	Maximum number of shards that the queue is allowed to run
		prometheus_remote_storage_shards_min	Minimum number of shards that the queue is allowed to run
		prometheus_tsdb_wal_segment_current	WAL segment index that TSDB is currently writing to
		prometheus_tsdb_head_chunks	Number of chunks in the head block
		prometheus_tsdb_head_series	Number of series in the head block
		prometheus_tsdb_head_samples_appended_total	Number of appended samples
		prometheus_wal_watcher_current_segment	Current segment the WAL watcher is reading records from
		prometheus_target_interval_length_seconds	Actual intervals between scrapes
		prometheus_target_interval_length_seconds_count	Actual intervals between scrapes (count)
		prometheus_target_interval_length_seconds_sum	Actual intervals between scrapes (sum)
		prometheus_target_scrapes_exceeded_body_size_limit_total	Number of scrapes that hit the body size limit
		prometheus_target_scrapes_exceeded_sample_limit_total	Number of scrapes that hit the sample limit
		prometheus_target_scrapes_sample_duplicate_timestamp_total	Number scraped samples with duplicate timestamps
		prometheus_target_scrapes_sample_out_of_bounds_total	Number of samples rejected due to timestamp falling outside of the time bounds

Target Name	Job Name	Metric	Description
		prometheus_target_scrapes_sample_out_of_order_total	Number of out-of-order samples
		prometheus_target_sync_length_seconds	Interval for synchronizing the scrape pool
		prometheus_target_sync_length_seconds_count	Interval for synchronizing the scrape pool (count)
		prometheus_target_sync_length_seconds_sum	Interval for synchronizing the scrape pool (sum)
		promhttp_metric_handler_requests_in_flight	Number of metrics being processed
		promhttp_metric_handler_requests_total	Number of metric processing times
		go_goroutines	Number of goroutines
podMonitor/monitoring/virtual-kubelet-pods/0	monitoring/virtual-kubelet-pods	container_cpu_load_average_10s	Value of container CPU load average over the last 10 seconds
		container_cpu_system_seconds_total	Cumulative container CPU system time
		container_cpu_usage_seconds_total	Cumulative CPU time consumed by a container in core-seconds
		container_cpu_user_seconds_total	Usage of user CPU time
		container_cpu_cfs_periods_total	Number of elapsed enforcement period intervals
		container_cpu_cfs_throttled_periods_total	Number of throttled period intervals
		container_cpu_cfs_throttled_seconds_total	Total time duration the container has been throttled
		container_fs_inodes_free	Number of available inodes in a file system

Target Name	Job Name	Metric	Description
		container_fs_usage_bytes	File system usage
		container_fs_inodes_total	Number of inodes in a file system
		container_fs_io_current	Number of I/Os currently in progress in a disk or file system
		container_fs_io_time_seconds_total	Cumulative seconds spent on doing I/Os by the disk or file system
		container_fs_io_time_weighted_seconds_total	Cumulative weighted I/O time of a disk or file system
		container_fs_limit_bytes	Total disk or file system capacity that can be consumed by a container
		container_fs_reads_bytes_total	Cumulative amount of disk or file system data read by a container
		container_fs_read_seconds_total	Cumulative number of seconds the container spent on reading disk or file system data
		container_fs_reads_merged_total	Cumulative number of merged disk or file system reads made by the container.
		container_fs_reads_total	Cumulative number of disk or file system reads completed by a container
		container_fs_sector_reads_total	Cumulative number of disk or file system sector reads completed by a container

Target Name	Job Name	Metric	Description
		container_fs_sector_writes_total	Cumulative number of disk or file system sector writes completed by a container
		container_fs_writes_bytes_total	Total amount of data written by a container to a disk or file system
		container_fs_write_seconds_total	Cumulative number of seconds the container spent on writing data to the disk or file system
		container_fs_writes_merged_total	Cumulative number of merged container writes to the disk or file system
		container_fs_writes_total	Cumulative number of disk or file system writes completed by a container
		container_blkio_device_usage_total	Blkio device bytes usage
		container_memory_failures_total	Cumulative number of container memory allocation failures
		container_memory_failcnt	Number of memory usage hits limits
		container_memory_cache	Memory used for the page cache of a container
		container_memory_mapped_file	Size of the container memory mapped file.
		container_memory_max_usage_bytes	Maximum memory usage recorded for a container
		container_memory_rss	Size of the resident memory set for a container
		container_memory_swap	Container swap usage

Target Name	Job Name	Metric	Description
		container_memory_usage_bytes	Current memory usage of a container
		container_memory_working_set_bytes	Memory usage of the working set of a container
		container_network_receive_bytes_total	Total volume of data received by the container network
		container_network_receive_errors_total	Cumulative number of errors encountered during reception
		container_network_receive_packets_dropped_total	Cumulative number of packets dropped during reception
		container_network_receive_packets_total	Cumulative number of packets received
		container_network_transmit_bytes_total	Total volume of data transmitted on the container network
		container_network_transmit_errors_total	Cumulative number of errors encountered during transmission
		container_network_transmit_packets_dropped_total	Cumulative number of packets dropped during transmission
		container_network_transmit_packets_total	Cumulative number of packets transmitted
		container_processes	Number of processes running inside the container
		container_sockets	Number of open sockets for the container
		container_file_descriptors	Number of open file descriptors for a container
		container_threads	Number of threads running inside the container

Target Name	Job Name	Metric	Description
		container_threads_max	Maximum number of threads allowed inside the container
		container_ulimits_soft	Soft ulimit value of process 1 in the container. Unlimited if the value is **-1**, except priority and nice.
		container_tasks_state	Number of tasks in the specified state, such as sleeping, running, stopped, uninterruptible, or ioawaiting
		container_spec_cpu_period	CPU period of the container
		container_spec_cpu_shares	CPU share of the container
		container_spec_cpu_quota	CPU quota of the container
		container_spec_memory_limit_bytes	Memory limit for the container
		container_spec_memory_reservation_limit_bytes	Memory reservation limit for the container
		container_spec_memory_swap_limit_bytes	Memory swap limit for the container
		container_start_time_seconds	Running time of the container.
		container_last_seen	Last time a container was seen by the exporter
		container_accelerator_memory_used_bytes	GPU accelerator memory that is being used by the container
		container_accelerator_memory_total_bytes	Total available memory of a GPU accelerator

Target Name	Job Name	Metric	Description
		container_accelerator_duty_cycle	Percentage of time when a GPU accelerator is actually running
podMonitor/monitoring/everest-csi-controller/0	monitoring/everest-csi-controller	everest_action_result_total	Number of action results
		everest_function_duration_seconds_bucket	Histogram of action duration (bucket)
		everest_function_duration_seconds_count	Histogram of action duration (count)
		everest_function_duration_seconds_sum	Histogram of action duration (sum)
		everest_function_duration_quantile_seconds	Time quantile required by the action
		node_volume_read_completed_total	Number of completed reads
		node_volume_read_merged_total	Number of merged reads
		node_volume_read_bytes_total	Total number of bytes read by a sector
		node_volume_read_time_milliseconds_total	Total read duration
		node_volume_write_completed_total	Number of completed writes
		node_volume_write_merged_total	Number of merged writes
		node_volume_write_bytes_total	Total number of bytes written into a sector
		node_volume_write_time_milliseconds_total	Total write duration
		node_volume_io_now	Number of ongoing I/Os
		node_volume_io_time_seconds_total	Total I/O operation duration
		node_volume_capacity_bytes_available	Available capacity
		node_volume_capacity_bytes_total	Total capacity

Target Name	Job Name	Metric	Description
		node_volume_capacity_bytes_used	Used capacity
		node_volume_inodes_available	Available inodes
		node_volume_inodes_total	Total number of inodes
		node_volume_inodes_used	Used inodes
		node_volume_read_transmissions_total	Number of read transmission times
		node_volume_read_timeouts_total	Number of read timeouts
		node_volume_read_sent_bytes_total	Number of bytes read
		node_volume_read_queue_time_milliseconds_total	Read queue waiting time
		node_volume_read_rtt_time_milliseconds_total	Read RTT
		node_volume_write_transmissions_total	Number of write transmissions
		node_volume_write_timeouts_total	Number of write timeouts
		node_volume_write_queue_time_milliseconds_total	Write queue waiting time
		node_volume_write_rtt_time_milliseconds_total	Write RTT
		node_volume_localvolume_stats_capacity_bytes	Local storage capacity
		node_volume_localvolume_stats_available_bytes	Available local storage
		node_volume_localvolume_stats_used_bytes	Used local storage

Target Name	Job Name	Metric	Description
		node_volume_localvolume_stats_inodes	Number of inodes for a local volume
		node_volume_localvolume_stats_inodes_used	Used inodes for a local volume
podMonitor/monitoring/nginx-ingress-controller/0	monitoring/nginx-ingress-controller	nginx_ingress_controller_bytes_sent	Number of bytes sent to the client
		nginx_ingress_controller_connect_duration_seconds	Duration for connecting to the upstream server
		nginx_ingress_controller_header_duration_seconds	Time required for receiving the first header from the upstream server
		nginx_ingress_controller_ingress_upstream_latency_seconds	Upstream service latency
		nginx_ingress_controller_request_duration_seconds	Time required for processing a request, in milliseconds
		nginx_ingress_controller_request_size	Length of a request, including the request line, header, and body
		nginx_ingress_controller_requests	Total number of HTTP requests processed by Nginx Ingress Controller since it starts
		nginx_ingress_controller_response_duration_seconds	Time required for receiving the response from the upstream server
		nginx_ingress_controller_response_size	Length of a response, including the request line, header, and body
		nginx_ingress_controller_nginx_process_connections	Number of client connections in the active, read, write, or wait state

Target Name	Job Name	Metric	Description
		nginx_ingress_controller_nginx_process_connections_total	Total number of client connections in the accepted or handled state
		nginx_ingress_controller_nginx_process_cpu_seconds_total	Total CPU time consumed by the Nginx process (unit: second)
		nginx_ingress_controller_nginx_process_num_procs	Number of processes
		nginx_ingress_controller_nginx_process_oldest_start_time_seconds	Start time in seconds since January 1, 1970
		nginx_ingress_controller_nginx_process_read_bytes_total	Number of bytes read
		nginx_ingress_controller_nginx_process_requests_total	Total number of requests processed by Nginx since startup
		nginx_ingress_controller_nginx_process_resident_memory_bytes	Resident memory usage of a process, that is, the actual physical memory usage
		nginx_ingress_controller_nginx_process_virtual_memory_bytes	Virtual memory usage of a process, that is, the total memory allocated to the process, including the actual physical memory and virtual swap space
		nginx_ingress_controller_nginx_process_write_bytes_total	Amount of data written by the Nginx process to disks or other devices for long-term storage
		nginx_ingress_controller_build_info	Build information of Nginx Ingress Controller, including the version and compilation time

Target Name	Job Name	Metric	Description
		nginx_ingress_controller_check_success	Health check result of Nginx Ingress Controller. **1**: Normal. **0**: Abnormal
		nginx_ingress_controller_config_hash	Configured hash value
		nginx_ingress_controller_config_last_reload_successful	Whether the Nginx Ingress Controller configuration is successfully reloaded
		nginx_ingress_controller_config_last_reload_successful_timestamp_seconds	Last timestamp when the Nginx Ingress Controller configuration was successfully reloaded
		nginx_ingress_controller_ssl_certificate_info	Nginx Ingress Controller certificate information
		nginx_ingress_controller_success	Cumulative number of reload operations of Nginx Ingress Controller
		nginx_ingress_controller_orphan_ingress	Whether the ingress is isolated. 1: Isolated. 0: Not isolated. **namespace** indicates the namespace where the ingress is located, **ingress** indicates the ingress name. **type** indicates that the isolation type (options: **no-service** and **no-endpoint**).
		nginx_ingress_controller_admission_config_size	Size of the admission controller configuration
		nginx_ingress_controller_admission_render_duration	Rendering duration of the admission controller
		nginx_ingress_controller_admission_render_ingresses	Length of ingresses rendered by the admission controller

Target Name	Job Name	Metric	Description
		nginx_ingress_controller_admission_roundtrip_duration	Time spent by the admission controller to process new events
		nginx_ingress_controller_admission_tested_duration	Time spent on admission controller tests
		nginx_ingress_controller_admission_tested_ingresses	Length of ingresses processed by the admission controller

8.4 Basic Metrics: ModelArts Metrics

This section describes the ModelArts metrics reported to AOM through the Agent.

Table 8-4 Metrics reported by ModelArts to AOM through the Agent

Category	Metric	Metric Name	Description	Value Range	Unit
CPU	ma_container_cpu_util	CPU Usage	CPU usage of a measured object	0–100	%
	ma_container_cpu_used_core	Used CPU Cores	Number of CPU cores used by a measured object	≥ 0	Cores
	ma_container_cpu_limit_core	Total CPU Cores	Total number of CPU cores that have been applied for a measured object	≥ 1	Cores
Memory	ma_container_memory_capacity_megabytes	Memory	Total physical memory that has been applied for a measured object	≥ 0	MB
	ma_container_memory_util	Physical Memory Usage	Percentage of the used physical memory to the total physical memory applied for a measured object	0–100	%

Category	Metric	Metric Name	Description	Value Range	Unit
	ma_container_memory_used_megabytes	Used Physical Memory	Physical memory that has been used by a measured object (**container_memory_working_set_bytes** in the current working set). (Memory usage in a working set = Active anonymous AND cache, and file-baked page ≤ **container_memory_usage_bytes**)	≥ 0	MB
Storage I/O	ma_container_disk_read_kilobytes	Disk Read Rate	Volume of data read from a disk per second	≥ 0	KB/s
	ma_container_disk_write_kilobytes	Disk Write Rate	Volume of data written into a disk per second	≥ 0	KB/s
GPU memory	ma_container_gpu_mem_total_megabytes	GPU Memory Capacity	Total GPU memory of a training job	> 0	MB
	ma_container_gpu_mem_util	GPU Memory Usage	Percentage of the used GPU memory to the total GPU memory	0–100	%
	ma_container_gpu_mem_used_megabytes	Used GPU Memory	GPU memory used by a measured object	≥ 0	MB
GPU	ma_container_gpu_util	GPU Usage	GPU usage of a measured object	0–100	%

Category	Metric	Metric Name	Description	Value Range	Unit
	ma_container_gpu_mem_copy_util	GPU Memory Bandwidth Usage	GPU memory bandwidth usage of a measured object. For example, the maximum memory bandwidth of NVIDIA GPU V100 is 900 GB/s. If the current memory bandwidth is 450 GB/s, the memory bandwidth usage is 50%.	0–100	%
	ma_container_gpu_enc_util	GPU Encoder Usage	GPU encoder usage of a measured object	0–100	%
	ma_container_gpu_dec_util	GPU Decoder Usage	GPU decoder usage of a measured object	0–100	%
	DCGM_FI_DEV_GPU_TEMP	GPU Temperature	GPU temperature	> 0	°C
	DCGM_FI_DEV_POWER_USAGE	GPU Power	GPU power	> 0	W
	DCGM_FI_DEV_MEMORY_TEMP	Memory Temperature	Memory temperature	> 0	°C

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_GR_ENGINE_ACTIVE	Graphics Engine Activity	Percentage of the time when the graphic or compute engine is in the active state within a period. This is an average value of all graphic or compute engines. An active graphic or compute engine indicates that the graphic or compute context is associated with a thread and the graphic or compute context is busy.	0–1.0	Percentage (fraction)
	DCGM_FI_PROF_SM_OCCUPANCY	SM Occupancy	Ratio of the number of thread bundles that reside on the SM to the maximum number of thread bundles that can reside on the SM within a period. This is an average value of all SMs within a period. A high value does not mean a high GPU usage. Only when the GPU memory bandwidth is limited, a high value of workloads (**DCGM_FI_PROF_DRAM_ACTIVE**) indicates more efficient GPU usage.	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_PIPE_TENSOR_ACTIVE	Tensor Activity	Fraction of the period during which the tensor (HMMA/IMMA) pipe is active. This is an average value within a period, not an instantaneous value. A higher value indicates a higher utilization of tensor cores. Value 1 (100%) indicates that a tensor instruction is sent every instruction cycle in the entire period (one instruction is completed in two cycles). If the value is 0.2 (20%), the possible causes are as follows: During the entire period, 20% of the SM tensor cores run at 100% utilization. During the entire period, all SM tensor cores run at 20% utilization. During 1/5 of the entire period, all SM tensor cores run at 100% utilization. Other combinations	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_DRAM_ACTIVE	Memory BW Utilization	Percentage of the time for sending data to or receiving data from the device memory within a period. This is an average value within a period, not an instantaneous value. A higher value indicates a higher utilization of device memory. Value 1 (100%) indicates that a DRAM instruction is executed once per cycle throughout a period (the maximum value can be reached at a peak of about 0.8). If the value is 0.2 (20%), indicating that data is read from or written into the device memory during 20% of the cycle within a period.	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_PIPE_FP16_ACTIVE	FP16 Engine Activity	Fraction of the period during which the FP16 (half-precision) pipe is active. This is an average value within a period, not an instantaneous value. A larger value indicates a higher usage of FP16 cores. Value 1 (100%) indicates that the FP16 instruction is executed every two cycles (for example, Volta cards) in a period. If the value is 0.2 (20%), the possible causes are as follows: During the entire period, 20% of the SM FP16 cores run at 100% utilization. During the entire period, all SM FP16 cores run at 20% utilization. During 1/5 of the entire period, all SM FP16 cores run at 100% utilization. Other combinations	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_PIPE_FP32_ACTIVE	FP32 Engine Activity	Fraction of the period during which the fused multiply-add (FMA) pipe is active. Multiply-add applies to FP32 (single precision) and integers. This is an average value within a period, not an instantaneous value. A larger value indicates a higher usage of FP32 cores. Value 1 (100%) indicates that the FP32 instruction is executed every two cycles (for example, Volta cards) in a period. If the value is 0.2 (20%), the possible causes are as follows: During the entire period, 20% of the SM FP32 cores run at 100% utilization. During the entire period, all SM FP32 cores run at 20% utilization. During 1/5 of the entire period, all SM FP32 cores run at 100% utilization. Other combinations	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_PIPE_FP64_ACTIVE	FP64 Engine Activity	Fraction of the period during which the FP64 (double precision) pipe is active. This is an average value within a period, not an instantaneous value. A larger value indicates a higher usage of FP64 cores. Value 1 (100%) indicates that the FP64 instruction is executed every four cycles (for example, Volta cards) in a period. If the value is 0.2 (20%), the possible causes are as follows: During the entire period, 20% of the SM FP64 cores run at 100% utilization. During the entire period, all SM FP64 cores run at 20% utilization. During 1/5 of the entire period, all SM FP64 cores run at 100% utilization. Other combinations	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_SM_ACTIVE	SM Activity	Fraction of the time during which at least one thread bundle is active on an SM within a period. This is an average value of all SMs and is insensitive to the number of threads in each block. A thread bundle is active after being scheduled and allocated with resources. The thread bundle may be in the computing state or a non-computing state (for example, waiting for a memory request). If the value is less than 0.5, GPUs are not efficiently used. The value should be greater than 0.8. For example, a GPU has N SMs: A kernel function uses N thread blocks to run on all SMs in a period. In this case, the value is 1 (100%). A kernel function runs N/5 thread blocks in a period. In this case, the value is 0.2. A kernel function uses N thread blocks and runs only 1/5 of cycles in a period. In this case, the value is 0.2.	0–1.0	Percentage (fraction)

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_PCIE_TX_BYTES DCGM_FI_PROF_PCIE_RX_BYTES	PCIe Bandwidth	Rate of data transmitted or received over the PCIe bus, including the protocol header and data payload. This is an average value within a period, not an instantaneous value. The rate is averaged over the period. For example, if 1 GB of data is transmitted within 1 second, the transmission rate is 1 GB/s regardless of whether the data is transmitted at a constant rate or burst. Theoretically, the maximum PCIe Gen3 bandwidth is 985 MB/s per channel.	≥ 0	Bytes/s

Category	Metric	Metric Name	Description	Value Range	Unit
	DCGM_FI_PROF_NVLINK_RX_BYTES DCGM_FI_PROF_NVLINK_TX_BYTES	NVLink Bandwidth	Rate at which data is transmitted or received through NVLink, excluding the protocol header. This is an average value within a period, not an instantaneous value. The rate is averaged over the period. For example, if 1 GB of data is transmitted within 1 second, the transmission rate is 1 GB/s regardless of whether the data is transmitted at a constant rate or burst. Theoretically, the maximum NVLink Gen2 bandwidth is 25 GB/s per link in each direction.	≥ 0	Bytes/s
Network I/O	ma_container_network_receive_bytes	Downlink Rate (BPS)	Inbound traffic rate of a measured object	≥ 0	Bytes/s
	ma_container_network_receive_packets	Downlink Rate (PPS)	Number of data packets received by a NIC per second	≥ 0	Packets/s
	ma_container_network_receive_error_packets	Downlink Error Rate	Number of error packets received by a NIC per second	≥ 0	Count/s
	ma_container_network_transmit_bytes	Uplink Rate (BPS)	Outbound traffic rate of a measured object	≥ 0	Bytes/s

Category	Metric	Metric Name	Description	Value Range	Unit
	ma_container_network_transmit_error_packets	Uplink Error Rate	Number of error packets sent by a NIC per second	≥ 0	Count/s
	ma_container_network_transmit_packets	Uplink Rate (PPS)	Number of data packets sent by a NIC per second	≥ 0	Packets/s
NPU	ma_container_npu_util	NPU Usage	NPU usage of a measured object	0–100	%
	ma_container_npu_memory_util	NPU Memory Usage	Percentage of the used NPU memory to the total NPU memory	0–100	%
	ma_container_npu_memory_used_megabytes	Used NPU Memory	NPU memory used by a measured object	≥ 0	MB
	ma_container_npu_memory_total_megabytes	Total NPU Memory	Total NPU memory of a measured object	≥ 0	MB

8.5 Basic Metrics: CSE Metrics

This section describes the types, names, and meanings of Cloud Service Engine (CSE) metrics reported to AOM.

Table 8-5 CSE metrics

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
ServiceComb	registry	servicecomb_service_center_db_service_total	Microservice Versions	Number of microservice versions	≥ 0	Count
		servicecomb_service_center_db_instance_total	Microservice Instances	Number of microservice instances	≥ 0	Count
		servicecomb_service_center_http_request_total	HTTP Requests	Number of HTTP requests, covering multiple URLs, methods, and codes	≥ 0	Count
		servicecomb_service_center_http_request_durations_microseconds	Total HTTP Request Time	Total HTTP request time, covering multiple URLs, methods, and codes	≥ 0	μs
	config	servicecomb_kie_request_count	HTTP Requests	Number of HTTP requests, covering multiple URLs, methods, and codes	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		servicecomb_kie_request_process_duration	Total HTTP Request Time	Total HTTP request time, covering multiple URLs, methods, and codes	≥ 0	ms
		servicecomb_kie_config_count	Configs	Number of ServiceComb configs	≥ 0	Count
Nacos	config	nacos_configCount	Nacos Configs	Number of configs in each Nacos cluster node	≥ 0	Count
		nacos_getConfig	Nacos Config Read Requests	Number of config read requests in each Nacos cluster node	≥ 0	Count
		nacos_longPolling	HTTP Persistent Connections of Nacos Config (Listeners)	Number of HTTP persistent connections of Nacos config	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		nacos_publish	Nacos Config Write Requests	Number of config write requests in each Nacos cluster node	≥ 0	Count
		nacos_subscriberCount	Nacos Config Subscribers	Number of Nacos config subscribers	≥ 0	Count
		nacos_configPushCost	Nacos Config Push Time	Nacos config push time	≥ 0	ms
	http	nacos_http_server_requests_seconds_count	HTTP Requests	Number of HTTP requests, covering multiple URLs, methods, and codes	≥ 0	Count
		nacos_http_server_requests_seconds_max	Maximum HTTP Request Time	Maximum HTTP request time, covering URLs, methods, and codes. This parameter is reported when Nacos-Client 1.x is used.	≥ 0	s

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		nacos_http_server_requests_seconds_sum	Total HTTP Request Time	Total HTTP request time, covering multiple URLs, methods, and codes	≥ 0	s
	naming	nacos_avgPushCost	Avg. Nacos Naming Push Time	Average Nacos naming push time	≥ 0	ms
		nacos_maxPushCost	Max. Nacos Naming Push Time	Maximum Nacos naming push time	≥ 0	ms
		nacos_failedPush	Nacos Naming Push Failures	Number of Nacos naming push failures	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		nacos_ipCount	Nacos Naming IP Addresses	Number of microservice instances that are registered	≥ 0	Count
		nacos_serviceSubscriberCount	Nacos Naming Subscribers	Number of Nacos naming subscribers	≥ 0	Count
		nacos_serviceCount	Nacos Naming Domain Names (2.x)	Number of services in each Nacos cluster node	≥ 0	Count
Application gateway	envoy	cpuUsage	CPU Usage	CPU usage of a measured object	0–100	%
		envoy_http_downstream_cx_active	Active Connections	Number of active connections	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		downstream_cx_delayed_close_timeout	Connections Delayed to Close	Number of connections that are delayed to close	≥ 0	Count
		envoy_http_downstream_cx_destroy	Destroyed Connections	Number of connections that are destroyed	≥ 0	Count
		envoy_http_downstream_cx_destroy_active_rq	Destroyed Active Connections	Number of active connections that are destroyed	≥ 0	Count
		envoy_http_downstream_cx_destroy_local	Destroyed Local Connections	Number of local connections that are destroyed	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_http_downstream_cx_destroy_local_active_rq	Destroyed Local Active Connections	Number of local active connections that are destroyed	≥ 0	Count
		envoy_http_downstream_cx_destroy_remote	Destroyed Connections Due to Remote Shutdown	Number of connections that are destroyed due to remote shutdown	≥ 0	Count
		envoy_http_downstream_cx_destroy_remote_active_rq	Destroyed Active Connections Due to Remote Shutdown	Number of active connections that are destroyed due to remote shutdown	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_http_downstream_cx_drain_close	Closed Connections Due to Ejections	Number of connections that are closed due to ejections	≥ 0	Count
		envoy_http_downstream_cx_http1_active	HTTP1 Connections	Number of HTTP1 connections	≥ 0	Count
		envoy_http_downstream_cx_max_duration_reached	Timeout Connections	Number of connections that timed out	≥ 0	Count
		envoy_http_downstream_cx_tx_bytes_total	Total Sent Bytes	Total number of bytes that are sent	≥ 0	Byte
		envoy_http_downstream_rq	Total Requests	Total number of requests	≥ 0	Count
		envoy_http_downstream_rq_http1_total	Total HTTP1 Requests	Total number of HTTP1 requests	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_http_downstream_rq_http2_total	Total HTTP2 Requests	Total number of HTTP2 requests	≥ 0	Count
		envoy_http_downstream_rq_idle_timeout	Closed Requests Due to Excessive Idle Time	Number of requests that are closed due to excessive idle time	≥ 0	Count
		envoy_http_downstream_rq_too_large	Requests with Too Large Bodies	Number of requests with too large bodies (status code **413** returned)	≥ 0	Count
		downstream_rq_ws_on_non_ws_route	WebSocket Requests Without Routes	Number of requests that are rejected due to a lack of routes	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_http_local_rate_limiter_http_local_rate_limit_enforced	Limited Requests	Number of requests that are limited	≥ 0	Count
		envoy_cluster_circuit_breakers_default_cx_open	Connection Circuit Breaker Trigger Status	0: The concurrency limit has not been reached. 1: The concurrency limit has been reached. No more requests can be accepted.	0 or 1	N/A
		envoy_cluster_circuit_breakers_high_cx_open				
		envoy_cluster_circuit_breakers_default_cx_pool_open	Pool's Circuit Breaker Trigger Status	0: The concurrency limit has not been reached. 1: The concurrency limit has been reached. No more requests can be accepted.	0 or 1	N/A
		envoy_cluster_circuit_breakers_high_cx_pool_open				
		envoy_cluster_circuit_breakers_default_remaining_cx	Remaining Connections	Number of remaining connections that can be accepted by the connection circuit breaker	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_circuit_breakers_high_remaining_cx				
		envoy_cluster_circuit_breakers_default_remaining_cx_pools	Pool's Remaining Connections	Number of remaining connections that can be accepted by the pool circuit breaker	≥ 0	Count
		envoy_cluster_circuit_breakers_high_remaining_cx_pools				
		envoy_cluster_circuit_breakers_default_remaining_pending	Pending Requests	Number of requests to be processed before the circuit breaker reaches the concurrency limit	≥ 0	Count
		envoy_cluster_circuit_breakers_high_remaining_pending				

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_circuit_breakers_default_remaining_retries	Remaining Retries	Number of remaining retries before the circuit breaker reaches the concurrency limit	≥ 0	Count
		envoy_cluster_circuit_breakers_high_remaining_retries				
		envoy_cluster_circuit_breakers_default_remaining_rq	Remaining Requests	Number of remaining requests before the circuit breaker reaches the concurrency limit	≥ 0	Count
		envoy_cluster_circuit_breakers_high_remaining_rq				
		envoy_cluster_circuit_breakers_default_rq_open	Request Circuit Breaker Trigger Status	0: The concurrency limit has not been reached. 1: The concurrency limit has been reached. No more requests can be accepted.	0 or 1	N/A
		envoy_cluster_circuit_breakers_high_rq_open				

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_circuit_breakers_default_rq_retry_open	Retry Circuit Breaker Trigger Status	0: The concurrency limit has not been reached. 1: The concurrency limit has been reached. No more requests can be accepted.	0 or 1	N/A
		envoy_cluster_circuit_breakers_high_rq_retry_open				
		envoy_cluster_ejections_overflow	Ejections Due to Overflow	Number of ejections occurred due to overflow	≥ 0	Count
		envoy_cluster_ejections_consecutive_5xx	Ejections Caused by Consecutive 5xx Errors	Number of ejections that are caused by consecutive 5xx errors	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_detected_consecutive_5xx	Detected Ejections Caused by Consecutive 5xx Errors	Number of detected ejections (even if not forcibly enforced) that are caused by consecutive 5xx errors	≥ 0	Count
		envoy_cluster_ejections_detected_consecutive_gateway_failure	Detected Ejections Caused by Consecutive Gateway Faults	Number of detected ejections (even if not forcibly enforced) that are caused by consecutive gateway faults	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_detected_consecutive_local_origin_failure	Detected Ejections Caused by Consecutive Local Faults	Number of detected ejections (even if not forcibly enforced) that are caused by consecutive local faults	≥ 0	Count
		envoy_cluster_ejections_enforced_consecutive_local_origin_failure	Forced Ejections Caused by Consecutive Local Faults	Number of forced ejections that are caused by consecutive local faults	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_detected_failure_percentage	Ejections Caused by High Request Failure Rate	Number of ejections occurred because the request failure rate exceeds the threshold	≥ 0	Count
		envoy_cluster_ejections_detected_local_origin_failure_percentage	Detected ejections Caused by High Local Request Failure Rate	Number of ejections occurred because the local request failure rate exceeds the threshold	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_detected_local_origin_success_rate	Detected Ejections Caused by Low Local Request Success Rate	Number of ejections occurred (even if not forcibly enforced) because the local request success rate does not reach the threshold	≥ 0	Count
		envoy_cluster_ejections_detected_success_rate	Detected Ejections Caused by Low Request Success Rate	Number of ejections occurred because the request success rate does not reach the threshold	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_enforced_consecutive_5xx	Enforced Ejections Caused by Consecutive 5xx Errors	Number of forced ejections that are caused by consecutive 5xx errors	≥ 0	Count
		envoy_cluster_ejections_enforced_consecutive_gateway_failure	Forced Ejections Caused by Consecutive Gateway Faults	Number of forced ejections that are caused by consecutive gateway faults	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_enforced_failure_percentage	Forced Ejections Caused by High Request Failure Rate	Number of forced ejections occurred because the request failure rate exceeds the threshold	≥ 0	Count
		envoy_cluster_ejections_enforced_local_origin_failure_percentage	Forced Ejections Caused by High Local Request Failure Rate	Number of forced ejections occurred because the local request failure rate exceeds the threshold	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_cluster_ejections_enforced_local_origin_success_rate	Forced Ejections Caused by Low Local Request Success Rate	Number of forced ejections occurred because the local request success rate does not reach the threshold	≥ 0	Count
		envoy_cluster_ejections_enforced_success_rate	Forced Ejections Caused by Low Request Success Rate	Number of forced ejections occurred because the request success rate does not reach the threshold	≥ 0	Count
		envoy_cluster_ejections_enforced_total	Forced Ejections	Number of forced ejections that are caused by any exception	≥ 0	Count

Category	Sub-Category	Metric	Metric Name	Description	Value Range	Unit
		envoy_http_downstream_cx_rx_bytes_total	Total Bytes Received	Total number of bytes that are received	≥ 0	Byte

8.6 Basic Metrics: Node Exporter Metrics

This section describes the types, names, and meanings of metrics reported by Node Exporter to AOM.

Table 8-6 Metrics of containers running in CCE or on-premises Kubernetes clusters

Job	Metric	Description
node-exporter	node_filesystem_size_bytes	Consumed space of a file system
	node_filesystem_readonly	Read-only file system
	node_filesystem_free_bytes	Remaining space of a file system
	node_filesystem_avail_bytes	File system space that is available for use
	node_cpu_seconds_total	Seconds each CPU spent doing each type of work
	node_network_receive_bytes_total	Total amount of received data
	node_network_receive_errs_total	Cumulative number of errors encountered during reception
	node_network_transmit_bytes_total	Total amount of transmitted data
	node_network_receive_packets_total	Cumulative number of packets received
	node_network_transmit_drop_total	Cumulative number of dropped packets during transmission

Job	Metric	Description
	node_network_transmit_errs_total	Cumulative number of errors encountered during transmission
	node_network_up	NIC status
	node_network_transmit_packets_total	Cumulative number of packets transmitted
	node_network_receive_drop_total	Cumulative number of packets dropped during reception
	go_gc_duration_seconds	This value is obtained by calling the **debug.ReadGCStats()** function. When this function is called, the **PauseQuantile** field of the **GCStats** structure is set to **5**. In this way, the function returns 5 GC pause time percentiles (the minimum percentile, 25%, 50%, 75%, and maximum percentile). Then, the Prometheus Go client creates a summary metric based on the returned GC pause time percentile, **NumGC**, and **PauseTotal**.
	node_load5	5-minute average CPU load
	node_filefd_allocated	Allocated file descriptors
	node_exporter_build_info	Node exporter build information
	node_disk_written_bytes_total	Number of bytes that are written
	node_disk_writes_completed_total	Number of writes completed
	node_disk_write_time_seconds_total	Number of seconds spent by all writes
	node_nf_conntrack_entries	Number of currently allocated flow entries for connection tracking
	node_nf_conntrack_entries_limit	Maximum size of a connection tracking table
	node_processes_max_processes	PID limit value
	node_processes_pids	Number of PIDs
	node_sockstat_TCP_alloc	Number of allocated TCP sockets
	node_sockstat_TCP_inuse	Number of TCP sockets in use

Job	Metric	Description
	node_sockstat_TCP_tw	Number of TCP sockets in the **TIME_WAIT** state
	node_timex_offset_seconds	Time offset
	node_timex_sync_status	Synchronization status of node clocks
	node_uname_info	Labeled system information as provided by the uname system call
	node_vmstat_pgfault	Number of page faults the system has made per second in **/proc/vmstat**
	node_vmstat_pgmajfault	Number of major faults per second in **/proc/vmstat**
	node_vmstat_pgpgin	Number of page in between main memory and block device in **/proc/vmstat**
	node_vmstat_pgpgout	Number of page out between main memory and block device in **/proc/vmstat**
	node_disk_reads_completed_total	Number of reads completed
	node_disk_read_time_seconds_total	Number of seconds spent by all reads
	process_cpu_seconds_total	The value is obtained based on the **utime** parameter (the number of ticks executed by the Go process in user mode) and the **stime** parameter (the number of ticks executed by the Go process in kernel mode, for example, during system invocation). Unit: jiffies, which measure the tick time between two system timer interruptions. process_cpu_seconds_total = (utime + stime)/USER_HZ Based on the preceding formula, you can obtain the total time (unit: seconds) for a process to run on the OS.
	node_disk_read_bytes_total	Number of bytes that are read

Job	Metric	Description
	node_disk_io_time_weighted_seconds_total	The weighted number of seconds spent doing I/Os
	node_disk_io_time_seconds_total	Total seconds spent doing I/Os
	node_disk_io_now	Number of I/Os in progress
	node_context_switches_total	Number of context switches
	node_boot_time_seconds	Node boot time
	process_resident_memory_bytes	Resident set size (RSS), which is the memory actually used by a process. It includes the shared memory, but does not include the allocated but unused memory or swapped-out memory.
	node_intr_total	Number of interruptions that occurred
	node_load1	1-minute average CPU load
	go_goroutines	This value is obtained by calling **runtime.NumGoroutine()** and calculated based on the **sched** scheduler structure and global **allglen** variable. Fields in the **sched** structure may change concurrently. Therefore, the system checks whether the calculated value is less than **1**. If the value is less than **1**, the system returns **1**.
	scrape_duration_seconds	Time spent on collecting information about the scrape target
	node_load15	15-minute average CPU load
	scrape_samples_post_metric_relabeling	Number of remaining samples after metrics are relabeled
	node_netstat_Tcp_PassiveOpens	Number of TCP connections that directly change from the **LISTEN** state to the **SYN-RCVD** state
	scrape_samples_scraped	Number of samples scraped
	node_netstat_Tcp_CurrEstab	Number of TCP connections in the **ESTABLISHED** or **CLOSE-WAIT** state

Job	Metric	Description
	scrape_series_added	Number of series added to the scrape target
	node_netstat_Tcp_ActiveOpens	Number of TCP connections that directly change from the **CLOSED** state to the **SYN-SENT** state
	node_memory_MemTotal_bytes	Total memory of a node
	node_memory_MemFree_bytes	Free memory of a node
	node_memory_MemAvailable_bytes	Available memory of a node
	node_memory_Cached_bytes	Memory for the node page cache
	up	Scrape target status
	node_memory_Buffers_bytes	Memory of the node buffer

8.7 Basic Metrics: Flink Metrics

This section describes the categories, names, and meanings of Flink metrics reported to AOM.

Table 8-7 Flink metrics

Category	Metric	Description	Unit
CPU	flink_jobmanager_Status_JVM_CPU_Load	CPU load of the JVM in JobManager	N/A
	flink_jobmanager_Status_JVM_CPU_Time	CPU time of the JVM in JobManager	N/A
	flink_jobmanager_Status_ProcessTree_CPU_Usage	CPU usage of the JVM in JobManager	N/A
	flink_taskmanager_Status_JVM_CPU_Load	CPU load of the JVM in TaskManager	N/A
	flink_taskmanager_Status_JVM_CPU_Time	CPU time of the JVM in TaskManager	N/A
	flink_taskmanager_Status_ProcessTree_CPU_Usage	CPU usage of the JVM in TaskManager	N/A

Category	Metric	Description	Unit
Memory	flink_jobmanager_Status_JVM_Memory_Heap_Used	Used heap memory of JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_Heap_Committed	Available JVM heap memory of JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_Heap_Max	Maximum heap memory that can be used for memory management in JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_NonHeap_Used	Used off-heap memory of JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_NonHeap_Committed	Available JVM off-heap memory of JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_NonHeap_Max	Maximum off-heap memory that can be used for memory management in JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_Metaspace_Used	Used memory of the JobManager MetaSpace memory pool	Bytes
	flink_jobmanager_Status_JVM_Memory_Metaspace_Committed	Available JVM memory of the JobManager MetaSpace memory pool	Bytes
	flink_jobmanager_Status_JVM_Memory_Metaspace_Max	Maximum memory that can be used in the JobManager MetaSpace memory pool	Bytes
	flink_jobmanager_Status_JVM_Memory_Direct_Count	Number of buffers in the direct buffer pool of JobManager	N/A
	flink_jobmanager_Status_JVM_Memory_Direct_MemoryUsed	Memory for the direct buffer pool in JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_Direct_TotalCapacity	Total capacity of all buffers in the direct buffer pool of JobManager	Bytes

Category	Metric	Description	Unit
	flink_jobmanager_Status_JVM_Memory_Mapped_Count	Number of buffers in the mapped buffer pool of JobManager	N/A
	flink_jobmanager_Status_JVM_Memory_Mapped_MemoryUsed	Memory for the mapped buffer pool in JobManager	Bytes
	flink_jobmanager_Status_JVM_Memory_Mapped_TotalCapacity	Total capacity of all buffers in the mapped buffer pool of JobManager	Bytes
	flink_jobmanager_Status_Flink_Memory_Managed_Used	Managed memory that has been used in JobManager	Bytes
	flink_jobmanager_Status_Flink_Memory_Managed_Total	Total managed memory of JobManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Heap_Used	Used heap memory of TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Heap_Committed	Available JVM heap memory of TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Heap_Max	Maximum heap memory that can be used for memory management in TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_NonHeap_Used	Off-heap memory of TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_NonHeap_Committed	Available JVM off-heap memory of TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_NonHeap_Max	Maximum off-heap memory that can be used for memory management in TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Metaspace_Used	Used memory of the TaskManager MetaSpace memory pool	Bytes

Category	Metric	Description	Unit
	flink_taskmanager_Status_JVM_Memory_Metaspace_Committed	Available JVM memory of the TaskManager MetaSpace memory pool	Bytes
	flink_taskmanager_Status_JVM_Memory_Metaspace_Max	Maximum memory that can be used in the TaskManager MetaSpace memory pool	Bytes
	flink_taskmanager_Status_JVM_Memory_Direct_Count	Number of buffers in the direct buffer pool of TaskManager	N/A
	flink_taskmanager_Status_JVM_Memory_Direct_MemoryUsed	Memory for the direct buffer pool in TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Direct_TotalCapacity	Total capacity of all buffers in the direct buffer pool of TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Mapped_Count	Number of buffers in the mapped buffer pool of TaskManager	N/A
	flink_taskmanager_Status_JVM_Memory_Mapped_MemoryUsed	Memory for the mapped buffer pool in TaskManager	Bytes
	flink_taskmanager_Status_JVM_Memory_Mapped_TotalCapacity	Total capacity of all buffers in the mapped buffer pool of TaskManager	Bytes
	flink_taskmanager_Status_Flink_Memory_Managed_Used	Managed memory that has been used in TaskManager	Bytes
	flink_taskmanager_Status_Flink_Memory_Managed_Total	Total managed memory of TaskManager	Bytes
	flink_taskmanager_Status_ProcessTree_Memory_RSS	Memory of the whole process in the Linux system	Bytes
Threads	flink_jobmanager_Status_JVM_Threads_Count	Total number of active threads in JobManager	Count
	flink_taskmanager_Status_JVM_Threads_Count	Total number of active threads in TaskManager	Count

Category	Metric	Description	Unit
Garbage collection	flink_jobmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Count	Number of garbage collection (GC) times of the JobManager Concurrent Mark Sweep (CMS) collector	Count
	flink_jobmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Time	Total time required for the JobManager CMS collector to collect garbage	ms
	flink_jobmanager_Status_JVM_GarbageCollector_ParNew_Count	Number of JobManager GC times	Count
	flink_jobmanager_Status_JVM_GarbageCollector_ParNew_Time	Each GC duration of JobManager	ms
	flink_taskmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Count	Number of GC times of the TaskManager CMS collector	Count
	flink_taskmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Time	Total time required for the TaskManager CMS collector to collect garbage	ms
	flink_taskmanager_Status_JVM_GarbageCollector_ParNew_Count	Number of TaskManager GC times	Count
	flink_taskmanager_Status_JVM_GarbageCollector_ParNew_Time	Each GC duration of TaskManager	ms
Class loader	flink_jobmanager_Status_JVM_ClassLoader_ClassesLoaded	Total number of classes that JobManager has loaded since the JVM started	N/A
	flink_jobmanager_Status_JVM_ClassLoader_ClassesUnloaded	Total number of classes that JobManager has unloaded since the JVM started	N/A
	flink_taskmanager_Status_JVM_ClassLoader_ClassesLoaded	Total number of classes that TaskManager has loaded since the JVM started	N/A

Category	Metric	Description	Unit
	flink_taskmanager_Status_JVM_ClassLoader_ClassesUnloaded	Total number of classes that TaskManager has unloaded since the JVM started	N/A
Network	flink_taskmanager_Status_Network_AvailableMemorySegments	Number of unused memory segments of TaskManager	N/A
	flink_taskmanager_Status_Network_TotalMemorySegments	Total number of allocated memory segments of TaskManager	N/A
Default shuffle service	flink_taskmanager_Status_Shuffle_Netty_AvailableMemorySegments	Number of unused memory segments of TaskManager	N/A
	flink_taskmanager_Status_Shuffle_Netty_UsedMemorySegments	Number of used memory segments of TaskManager	N/A
	flink_taskmanager_Status_Shuffle_Netty_TotalMemorySegments	Number of allocated memory segments of TaskManager	N/A
	flink_taskmanager_Status_Shuffle_Netty_AvailableMemory	Unused memory of TaskManager	Bytes
	flink_taskmanager_Status_Shuffle_Netty_UsedMemory	Used memory of TaskManager	Bytes
	flink_taskmanager_Status_Shuffle_Netty_TotalMemory	Allocated memory of TaskManager	Bytes
Availability	flink_jobmanager_job_numRestarts	Total number of restarts since job submission	Count
Checkpoint	flink_jobmanager_job_lastCheckpointDuration	Time required to complete the latest checkpoint	ms
	flink_jobmanager_job_lastCheckpointSize	Size of the latest checkpoint. If incremental checkpoints are enabled or logs are changed, this metric may be different from **lastCheckpointFullSize**.	Bytes

Category	Metric	Description	Unit
	flink_jobmanager_job_numberOfInProgressCheck-points	Number of checkpoints that are in progress	Count
	flink_jobmanager_job_numberOfCompletedCheck-points	Number of checkpoints that are completed	Count
	flink_jobmanager_job_numberOfFailedCheckpoints	Number of failed checkpoints	Count
	flink_jobmanager_job_totalNumberOfCheckpoints	Total number of checkpoints	Count
I/O	flink_taskmanager_job_task_numBytesOut	Total number of bytes output by a task	Bytes
	flink_taskmanager_job_task_numBytesOutPerSecond	Total number of bytes output by a task per second	Bytes/s
	flink_taskmanager_job_task_isBackPressured	Whether a backpressure event occurs	N/A
	flink_taskmanager_job_task_numRecordsIn	Total number of records received by a task	Count
	flink_taskmanager_job_task_numRecordsInPerSecond	Total number of records received by a task per second	Records/s
	flink_taskmanager_job_task_numBytesIn	Number of bytes received by a task	Bytes
	flink_taskmanager_job_task_numBytesInPerSecond	Number of bytes received by a task per second	Bytes/s
	flink_taskmanager_job_task_numRecordsOut	Total number of records sent by a task	Count
	flink_taskmanager_job_task_numRecordsOutPerSecond	Total number of records sent by a task per second	Records/s
	flink_taskmanager_job_task_operator_numRecordsIn	Total number of records received by an operator	Count
	flink_taskmanager_job_task_operator_numRecordsInPerSecond	Total number of records received by an operator per second	Records/s
	flink_taskmanager_job_task_operator_numRecordsOut	Total number of records sent by an operator	Count

Category	Metric	Description	Unit
	flink_taskmanager_job_task_operator_numRecordsOutPerSecond	Total number of records sent by an operator per second	Records/s
	flink_taskmanager_job_task_operator_sourceIdleTime	Idle duration at the source end	ms
	flink_taskmanager_job_task_operator_source_numRecordsIn	Total number of records input to the source	Count
	flink_taskmanager_job_task_operator_sink_numRecordsOut	Total number of records output from the sink	Count
	flink_taskmanager_job_task_operator_source_numRecordsInPerSecond	Number of records input to the source per second	Records/s
	flink_taskmanager_job_task_operator_sink_numRecordsOutPerSecond	Number of records output from the sink per second	Records/s
Kafka connector	flink_taskmanager_job_task_operator_currentEmitEventTimeLag	Interval between the data event time and the time when the data leaves the source	ms
	flink_taskmanager_job_task_operator_currentFetchEventTimeLag	Interval between the data event time and the time when the data enters the source	ms
	flink_taskmanager_job_task_operator_pendingRecords	Number of data records that have not been pulled by the source	Count

8.8 Metric Dimensions

Dimensions of VM Metrics Reported by ICAgents

Table 8-8 Dimensions of VM metrics reported by ICAgents

Category	Metric Dimension	Description
Network metrics	clusterId	Cluster ID
	hostID	Host ID

Category	Metric Dimension	Description
	nameSpace	Cluster namespace
	netDevice	NIC name
	nodeIP	Host IP address
	nodeName	Host name
Disk metrics	clusterId	Cluster ID
	diskDevice	Disk name
	hostID	Host ID
	nameSpace	Cluster namespace
	nodeIP	Host IP address
	nodeName	Host name
Disk partition metrics	diskPartition	Partition disk
	diskPartitionType	Disk partition type
File system metrics	clusterId	Cluster ID
	clusterName	Cluster name
	fileSystem	File system
	hostID	Host ID
	mountPoint	Mount point
	nameSpace	Cluster namespace
	nodeIP	Host IP address
	nodeName	Host name
Host metrics	clusterId	Cluster ID
	clusterName	Cluster name
	gpuName	GPU name
	gpuID	GPU ID
	npuName	NPU name
	npuID	NPU ID
	hostID	Host ID
	nameSpace	Cluster namespace
	nodeIP	Host IP address

Category	Metric Dimension	Description
	hostName	Host name
Cluster metrics	clusterId	Cluster ID
	clusterName	Cluster name
	projectId	Project ID
Container metrics	appID	Service ID
	appName	Service name
	clusterId	Cluster ID
	clusterName	Cluster name
	containerID	Container ID
	containerName	Container name
	deploymentName	Workload name
	kind	Application type
	nameSpace	Cluster namespace
	podID	Instance ID
	podIP	Pod IP address
	podName	Instance name
	serviceID	Inventory ID
	nodename	Host name
	nodeIP	Host IP address
	virtualServiceName	Istio virtual service name
	gpuID	GPU ID
	npuName	NPU name
	npuID	NPU ID
Process metrics	appName	Service name
	clusterId	Cluster ID
	clusterName	Cluster name
	nameSpace	Cluster namespace
	processID	Process ID
	processName	Process name

Category	Metric Dimension	Description
	serviceID	Inventory ID

9 Security

9.1 Identity Authentication and Access Control

9.1.1 Access Control for AOM

Identity Authentication

Present your identity credential and undergo identity authentication no matter whether you access AOM through the console or by calling APIs. In addition, login protection and login authentication policies are provided to harden identity authentication security. Based on IAM, AOM supports three identity authentication modes: **Password Policy**, **Access Keys**, and **Temporary Access Key**. It also provides **Login Protection** and **Login Authentication Policy**.

Access Control

If you need to assign different permissions to employees in your enterprise to access your AOM resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, fine-grained permissions management, and access control. IAM helps you secure access to your Huawei Cloud resources. For details, see **11 Permissions Management**.

9.2 Data Protection

AOM takes different measures to keep data secure and reliable.

Table 9-1 AOM data protection methods and features

Method	Description
Transmission encryption (HTTPS)	AOM supports HTTPS to enhance data transmission security.
Data redundancy	Metric, alarm, and configuration data is stored in multiple copies to ensure data reliability.

9.3 Audit and Logs

Audit

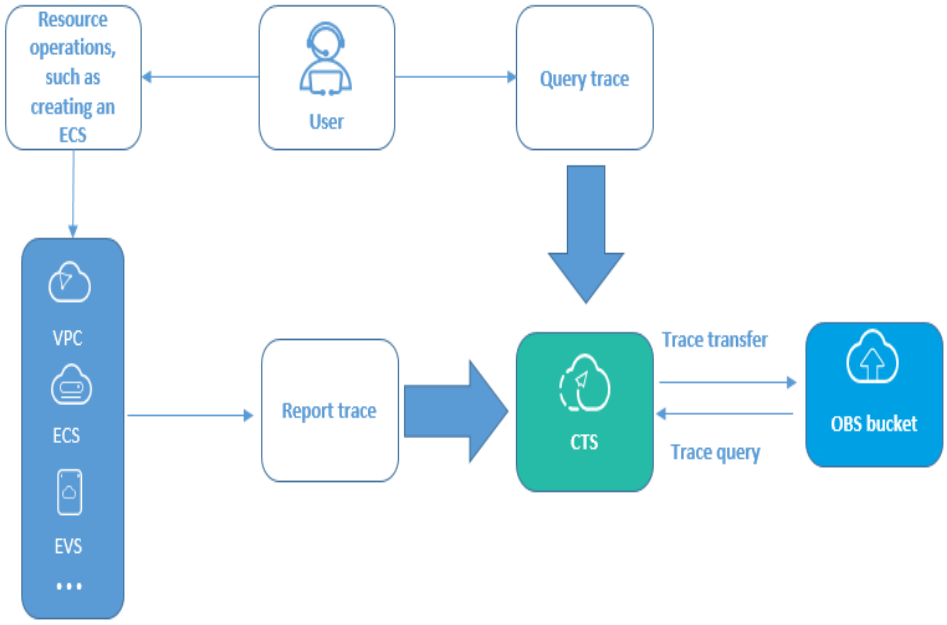
Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, trace resource changes, audit compliance, and locate faults.

After you enable CTS and configure a tracker, CTS records management traces of AOM for auditing.

For details about how to enable and configure CTS, see **Enabling CTS**.

For the management traces of AOM that can be recorded by CTS, see **Operations Logged by CTS**.

Figure 9-1 CTS



Logs

AOM collects container service logs and VM (ECS or BMS running Linux) logs and displays them on the console for you to search and view. For details, see **Log Analysis**.

9.4 Resilience

AOM provides multiple reliability DR capabilities. Technical solutions (such as intra-AZ instance DR, cross-AZ DR, cross-cluster DR, and multiple data copies) ensure service durability and reliability.

Table 9-2 Reliability architecture of AOM

Reliability Solution	Description
Intra-AZ instance DR	In a single AZ, multiple instances are used for DR. Faulty nodes can be quickly detected and remaining instances can still provide services.
Multi-AZ DR	AOM supports cross-AZ DR. When an AZ is abnormal, instances in other AZs can still provide services.
Cross-cluster DR	AOM supports cross-cluster DR. When one cluster is abnormal, AOM can continue to provide services.
Data DR	AOM configuration, metric, and alarm data is stored in multiple copies to ensure data reliability.

9.5 Security Risk Monitoring

AOM monitors security risks in various ways to ensure data security and reliability. For details, see **Table 9-3**.

Table 9-3 Monitoring security risks

Security Risk Monitoring	Description
Resource monitoring	AOM supports workload, cluster, and host monitoring, and metric browsing. It monitors your applications and cloud resources in real time and displays data in a visualized manner, helping you quickly analyze application health.
Alarm management	AOM allows you to set alarm conditions for applications, resources, and services. When AOM or its external service is or may be abnormal, email, SMS, or WeCom notifications will be sent to specified personnel.

10 Basic Concepts

10.1 Resource Monitoring

Table 10-1 Basic concepts

Terminology	Description
Metrics	Metrics reflect resource performance data or status. A metric consists of a namespace, dimension, name, and unit. Metric namespaces can be regarded as containers for storing metrics. Metrics in different namespaces are independent of each other so that metrics of different applications will not be aggregated to the same statistics information. Each metric has certain features, and a dimension may be considered as a category of such features.
Host	Each host of AOM corresponds to a VM or physical machine. A host can be your own VM or physical machine, or an Elastic Cloud Service (ECS) or Bare Metal Server (BMS) purchased. A host can be connected to AOM for monitoring only when its OS meets requirements and it is installed with an ICAgent.
Logs	AOM supports log collection, search, analysis, download, and dump. It also reports alarms based on keyword statistics and enables you to export reports, query SQL statements, and monitor data in real time. The log storage duration, size, and billing mode vary according to AOM editions..
Log traffic	Log traffic refers to the volume of logs reported per second. A maximum of 10 MB/s is supported for each tenant in a region. If the log traffic exceeds 10 MB/s, logs may be lost.
Alarms	Alarms are reported when AOM, ServiceStage, or CCE is abnormal or may cause exceptions. Alarms will cause service exceptions and need to be handled.

Terminology	Description
Events	Events generally carry important information. They are reported when AOM, ServiceStage, or CCE encounters some changes. Events do not necessarily cause service exceptions. Events do not need to be handled.
Alarm clearance	There are two alarm clearance modes: • Automatic clearance: After a fault is rectified, AOM automatically clears the corresponding alarm. • Manual clearance: After a fault is rectified, AOM does not automatically clear the corresponding alarm. Instead, you need to manually clear the alarm.
Alarm rules	Alarm rules are classified into metric alarm rules and event alarm rules. • Metric alarm rules monitor the usage of resources (such as hosts and components) in the environment in real time. • If there are many resource alarms but you do not want to receive notifications too often, set event alarm rules to quickly identify specific types of resource usage problems.
Alarm notification	There are two alarm notification modes: • Direct alarm reporting: When setting alarm notification rules, specify alarm notification recipients so that they can take measures to rectify faults in a timely manner. Alarms can be sent through email, DingTalk, WeCom, voice calls, and SMS. • Alarm noise reduction: Select a grouping rule to reduce alarm noise.
Alarm action rules	An alarm action rule defines the action to be taken after an alarm is generated. It includes where the message is sent and in what form. You can specify a message destination by setting **an SMN topic**.
Prometheus instances	Logical units used to manage Prometheus data collection, storage, and analysis.
Prometheus probes	Deployed in the Kubernetes clusters on the user or cloud product side. Prometheus probes automatically discover targets, collect metrics, and remotely write data to databases.
Exporters	Collect monitoring data and regulate the data provided for external systems using the Prometheus monitoring function. Currently, hundreds of official or third-party exporters are available. For details, see **Exporters**.
Jobs	Configuration set for a group of targets. Jobs specify the capture interval, access limit, and other behavior for a group of targets.

10.2 Collection Management

Table 10-2 Basic concepts of collection management

Terminology	Description
UniAgent	UniAgent manages the life cycle of plug-ins centrally and deliver instructions for operations such as script delivery or execution. It does not collect O&M data; instead, different plug-ins do so. Install, upgrade, and uninstall these plug-ins as required. More plug-ins (such as Cloud Eye and Host Security Service (HSS)) are coming soon.
AK/SK	Access key. You can install ICAgents using tenant-level AK/SK for easy log collection.
ICAgent	ICAgents collect metrics, logs, and application performance data. For the hosts purchased on the ECS or BMS console, manually install ICAgents. For the hosts that are purchased through CCE, ICAgents are automatically installed.
Installation host	You can deliver UniAgent installation instructions to hosts in batches through an installation host on AOM. After setting an installation host, you can remotely install UniAgents on other hosts in the same VPC.
Proxy area/ Proxy	To enable network communication between multiple clouds, purchase and configure an ECS as a proxy and bind an EIP to it. AOM delivers deployment and control instructions to remote hosts and receives O&M data through the proxy. A proxy area contains multiple proxies for high availability.

11 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your AOM resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your AOM resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users to control their access to specific types of resources. For example, some software developers in your enterprise need to use AOM resources but are not allowed to delete them or perform any high-risk operations such as deleting application discovery rules. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using AOM resources.

If your account does not need individual IAM users for permissions management, you may skip over this chapter.

IAM can be used free of charge. You pay only for the resources in your account. For more information, see **IAM Service Overview**.

AOM Permissions

By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies or roles to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on AOM.

AOM is a project-level service deployed and accessed in specific physical regions. To assign AOM permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing AOM, the users need to switch to a region where they have been authorized to use this service.

You can grant users permissions by using roles and policies.

- Roles: A coarse-grained authorization mechanism provided by IAM to define permissions based on users' job responsibilities. This mechanism provides only a limited number of service-level roles for authorization. Huawei Cloud

services depend on each other. When you grant permissions using roles, you may also need to attach dependent roles. However, roles are not an ideal choice for fine-grained authorization and secure access control.

- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant Elastic Cloud Server (ECS) users only the permissions for managing a certain type of ECSs. Most policies define permissions based on APIs.

Table 11-1 lists all the system permissions supported by AOM.

Table 11-1 System permissions supported by AOM

Subservice Name	Policy Name	Description	Type	Dependent System Permissions
Monitoring center / collection management	AOM FullAccess	Administrator permissions for AOM 2.0. Users granted these permissions can operate and use AOM.	System-defined policy	CCE FullAccess and DMS ReadOnly Access
	AOM ReadOnlyAccess	Read-only permissions for AOM 2.0. Users granted these permissions can only view AOM data.	System-defined policy	CCE ReadOnly Access and DMS ReadOnly Access

Common Operations and System Permissions for Resource Monitoring

Table 11-2 lists the common operations supported by each system-defined policy of resource monitoring. Select policies as required.

Table 11-2 Common operations supported by each system-defined policy

Operation	AOM FullAccess	AOM ReadOnlyAccess
Creating an alarm rule	√	x
Modifying an alarm rule	√	x
Deleting an alarm rule	√	x
Creating an alarm template	√	x

Operation	AOM FullAccess	AOM ReadOnlyAccess
Modifying an alarm template	√	x
Deleting an alarm template	√	x
Creating an alarm action rule	√	x
Modifying an alarm action rule	√	x
Deleting an alarm action rule	√	x
Creating a message template	√	x
Modifying a message template	√	x
Deleting a message template	√	x
Creating a grouping rule	√	x
Modifying a grouping rule	√	x
Deleting a grouping rule	√	x
Creating a suppression rule	√	x
Modifying a suppression rule	√	x
Deleting a suppression rule	√	x
Creating a silence rule	√	x
Modifying a silence rule	√	x
Deleting a silence rule	√	x
Creating a dashboard	√	x
Modifying a dashboard	√	x
Deleting a dashboard	√	x
Creating a Prometheus instance	√	x
Modifying a Prometheus instance	√	x

Operation	AOM FullAccess	AOM ReadOnlyAccess
Deleting a Prometheus instance	√	x
Creating an application discovery rule	√	x
Modifying an application discovery rule	√	x
Deleting an application discovery rule	√	x
Subscribing to threshold alarms	√	x
Configuring a VM log collection path	√	x

Common Operations Supported by Each System-defined Policy of Collection Management

Table 11-3 lists the common operations supported by each system-defined policy of collection management. Select policies as required.

Table 11-3 Common operations supported by each system-defined policy of collection management

Operation	AOM FullAccess	AOM ReadOnlyAccess
Querying a proxy area	√	√
Editing a proxy area	√	x
Deleting a proxy area	√	x
Creating a proxy area	√	x
Querying all proxies in a proxy area	√	√
Querying all proxy areas	√	√
Querying the Agent installation result	√	√

Operation	AOM FullAccess	AOM ReadOnlyAccess
Obtaining the Agent installation command of a host	√	√
Obtaining the host heartbeat and checking whether the host is connected with the server	√	√
Uninstalling running Agents in batches	√	x
Querying the Agent home page	√	√
Testing the connectivity between the installation host and the target host	√	x
Installing Agents in batches	√	x
Obtaining the latest operation log of the Agent	√	√
Obtaining the list of versions that can be selected during Agent installation	√	√
Obtaining the list of all Agent versions under the current project ID	√	√
Deleting hosts with Agents installed	√	x
Querying Agent information based on the ECS ID	√	√

Operation	AOM FullAccess	AOM ReadOnlyAccess
Deleting a host with an Agent installed	√	x
Setting an installation host	√	x
Resetting installation host parameters	√	x
Querying the list of hosts that can be set to installation hosts	√	√
Querying the list of Agent installation hosts	√	√
Deleting an installation host	√	x
Upgrading Agents in batches	√	x
Querying historical task logs	√	√
Querying historical task details	√	√
Querying all historical tasks	√	√
Querying all execution statuses and task types	√	√
Querying the Agent execution statuses in historical task details	√	√
Modifying a proxy	√	x
Deleting a proxy	√	x
Setting a proxy	√	x

Operation	AOM FullAccess	AOM ReadOnlyAccess
Querying the list of hosts that can be set to proxies	√	√
Updating plug-ins in batches	√	x
Uninstalling plug-ins in batches	√	x
Installing plug-ins in batches	√	x
Querying historical task logs of a plug-in	√	√
Querying all plug-in execution records	√	√
Querying plug-in execution records based on the task ID	√	√
Querying the plug-in execution statuses in historical task details	√	√
Obtaining the plug-in list	√	√
Querying the plug-in version	√	√
Querying the list of supported plug-ins	√	√
Obtaining the CCE cluster list	√	√
Obtaining the Agent list of a CCE cluster	√	√
Installing ICAgent on a CCE cluster	√	x
Upgrading ICAgent for a CCE cluster	√	x

Operation	AOM FullAccess	AOM ReadOnlyAccess
Uninstalling ICAgent from a CCE cluster	√	x
Obtaining the CCE cluster list	√	√
Obtaining the list of hosts where the ICAgent has been installed	√	√
Installing ICAgent on CCE cluster hosts	√	x
Upgrading ICAgent on CCE cluster hosts	√	x
Uninstalling ICAgent from CCE cluster hosts	√	x

Fine-grained Permissions

To use a custom fine-grained policy, log in to IAM as the administrator and select fine-grained permissions of AOM as required. For details about fine-grained permissions of AOM, see **Table 11-4**.

Table 11-4 Fine-grained permissions of AOM

Permission	Description	Permission Dependency	Application Scenario
aom:alarm:put	Reporting an alarm	N/A	Reporting a custom alarm
aom:event2AlarmRule:create	Adding an event alarm rule		Adding an event alarm rule
aom:event2AlarmRule:set	Modifying an event alarm rule		Modifying an event alarm rule
aom:event2AlarmRule:delete	Deleting an event alarm rule		Deleting an event alarm rule

Permission	Description	Permission Dependency	Application Scenario
aom:event2AlarmRule:list	Querying all event alarm rules		Querying all event alarm rules
aom:actionRule:create	Adding an alarm action rule		Adding an alarm action rule
aom:actionRule:delete	Deleting an alarm action rule		Deleting an alarm action rule
aom:actionRule:list	Querying the alarm action rule list		Querying the alarm action rule list
aom:actionRule:update	Modifying an alarm action rule		Modifying an alarm action rule
aom:actionRule:get	Querying an alarm action rule by name		Querying an alarm action rule by name
aom:alarm:list	Obtaining the sent alarm content		Obtaining the sent alarm content
aom:alarmRule:create	Creating a threshold rule		Creating a threshold rule
aom:alarmRule:set	Modifying a threshold rule		Modifying a threshold rule
aom:alarmRule:get	Querying threshold rules		Querying all threshold rules or a single threshold rule by rule ID
aom:alarmRule:delete	Deleting a threshold rule		Deleting threshold rules in batches or a single threshold rule by rule ID
aom:discoveryRule:list	Querying application discovery rules		Querying existing application discovery rules
aom:discoveryRule:delete	Deleting an application discovery rule		Deleting an application discovery rule
aom:discoveryRule:set	Adding an application discovery rule		Adding an application discovery rule

Permission	Description	Permission Dependency	Application Scenario
aom:metric:list	Querying time series objects		Querying time series objects
aom:metric:list	Querying time series data		Querying time series data
aom:metric:get	Querying metrics		Querying metrics
aom:metric:get	Querying monitoring data		Querying monitoring data
aom:muteRule:delete	Deleting a silence rule	N/A	Deleting a silence rule
aom:muteRule:create	Adding a silence rule		Adding a silence rule
aom:muteRule:update	Modifying a silence rule		Modifying a silence rule
aom:muteRule:list	Querying the silence rule list		Querying the silence rule list

Roles/Policies Required by AOM Dependent Services

If an IAM user needs to view data or use functions on the AOM console, grant the **AOM FullAccess** or **AOM ReadOnlyAccess** policy to the user group to which the user belongs and then add the roles or policies required by AOM dependent services by referring to **Table 11-5**.

NOTE

When a user subscribes to AOM for the first time, AOM will automatically create a service agency. In addition to the **AOM FullAccess** permission, the user must be granted the **Security Administrator** permission.

Table 11-5 Roles/Policies required by AOM dependent services

Console Function	Dependent Service	Policy/Role Required
• Workload monitoring • Cluster monitoring • Prometheus for CCE	CCE	To use workload and cluster monitoring and Prometheus for CCE, you need to set the **CCE FullAccess** permission.

12 Privacy Statement

All O&M data will be displayed on the AOM console. Therefore, do not upload your privacy or sensitive data to AOM. If necessary, encrypt such data.

Collector Deployment

When you manually install the ICAgent on an Elastic Cloud Server (ECS), your AK/SK will be used as an input parameter in the installation command. To prevent privacy leakage, disable historical record collection before installing the ICAgent. After the ICAgent is installed, it will encrypt and store your AK/SK.

Container Monitoring

For Cloud Container Engine (CCE) container monitoring, the AOM collector (ICAgent) must run as a privileged container. Evaluate the security risks of the privileged container and identify your container service scenarios. For example, for a node that provides services through logical multi-tenant container sharing, use open-source tools such as Prometheus to monitor the services and do not use ICAgent.

